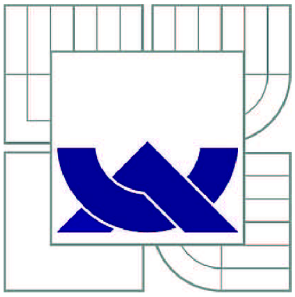
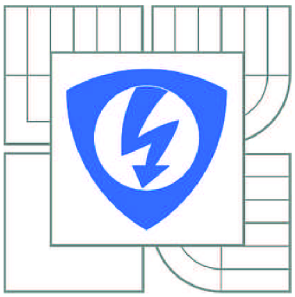


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

DEPARTMENT OF TELECOMMUNICATIONS

URČENÍ POZICE ÚTOČNÍKA PŘI POKUSU O NEOPRÁVNĚNÝ PŘÍSTUP DO OPERAČNÍHO SYSTÉMU

LOCATION OF ATTACKER ATTEMPTING UNAUTHORIZED ACCESS TO OPERATING SYSTEM

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

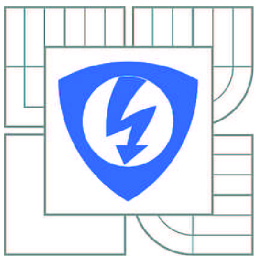
AUTOR PRÁCE — Bc. JOSEF POKORNÝ

AUTHOR

VEDOUCÍ PRÁCE — doc. Ing. DAN KOMOSNÝ, Ph.D.

SUPERVISOR

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student: Bc. Josef Pokorný ***ID:*** 47016
Ročník: 2 ***Akademický rok:*** 2012/2013

NÁZEV TÉMATU:

Určení pozice útočníka při pokusu o neoprávněný přístup do operačního systému

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s principy vyhodnocování fyzické polohy stanic v síti Internet. Zaměřte se na způsob zjištění pozice stanice pomocí veřejně dostupných pozičních databází. Realizujte aplikaci, která bude zjišťovat polohu potenciálního útočníka na operační systém. Pomocí reálných útoků ověřte správnou činnost navržené aplikace.

DOPORUČENÁ LITERATURA:

[1] POESE, I., UHLIG, S., KAAFAR, M., DONNET, B., GUEYE, B. IP Geolocation Databases: Unreliable? ACM SIGCOMM Computer Communication Review. ACM, 2011.
[2] MUIR, J., OORSCHOT, P.: Internet geolocation: Evasion and counterevasion. ACM Computing Surveys (CSUR). ACM, 2009.
[3] DOSTÁLEK L. et al.: Velký průvodce protokoly TCP/IP: Bezpečnost. 2. aktualizované vydání. Computer Press, 2003. ISBN 80-7226-849-X.

Termín zadání: 11.2.2013 ***Termín odevzdání:*** 29.5.2013

Vedoucí práce: doc. Ing. Dan Komosný, Ph.D.
Konzultanti diplomové práce:

prof. Ing. Kamil Vrba, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

ANOTACE

Předkládaná diplomová práce se zabývá určením fyzické polohy potenciálního útočníka na operační systém. Ve své práci zmiňuji základní způsoby útoku na operační systém: spam a viry, zjišťování informací o síti dostupných na internetu, skenování portů a detekce operačního systému. V práci popisuji užití programů: port skener Nmap, detektor skenování Scanlogd a prohlížeč systémových logů Swatch. Zabýval jsem se metodami určení fyzické polohy útočící stanice. Geolokační metody rozdělujeme na aktivní a pasivní. Aktivní metody spočívají v měření zpoždění v Internetu. Pasivní metody spočívají v dotazech do databáze na hledanou IP adresu. Popsal jsem volně dostupnou geolokační databázi Whois a databáze Maxmind. Vytvořil jsem aplikaci, která simuluje útok metodou skenování portů. Aplikace pracuje s datasetem reálných IP adres. Vytvořil jsem také aplikaci periodicky detekující útok. Skutečná poloha z datasetu a zjištěná poloha útočníka je potom zobrazena v mapě. Závěr práce se věnuje zhodnocení měření a porovnání naměřených dat s daty kolegů.

ABSTRACT

My master thesis estimates physical location of potential operating system attacker. It deals with basic methods of attack against an operating system: spam and viruses, searching the Internet, port scanning and operating system detection. The thesis disserts about a port scanner Nmap, a port scanning detector Scanlogd and about a system log watch Swatch. The thesis deals with geolocation methods of potential operating system attacker. These geolocation methods are divided into an active and a passive types. The active methods measure delay in the Internet. The passive methods query the database. I mentioned a freely accessible Whois database and MaxMind databases. There is a program developed and practically tested. The program simulates an attacker beginning an attack by scanning ports of target machine. The program works with dataset of real IP addresses. The program also detects the attack against operating system. The real and evaluated geographic location of an attacker is got and then shown in a map. At the end there is a review of results and data comparison with colleagues.

KLÍČOVÁ SLOVA

Fyzická poloha IP adresy, geolokační databáze MaxMind, útočník na operační systém, skener portů Nmap, simulátor útoku, detektor útoku.

KEYWORDS

Physical location of IP address, geolocation database MaxMind, operating system attacker, port scanner Nmap, attack simulator, attack detector.

POKORNÝ, J.*Určení pozice útočníka při pokusu o neoprávněný přístup do operačního systému.* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2013. 51 s. Vedoucí diplomové práce: doc. Ing. Dan Komosný, Ph.D..

Prohlášení o původnosti práce:

Prohlašuji, že svou diplomovou práci na téma „*Určení pozice útočníka při pokusu o neoprávněný přístup do operačního systému*“ jsem vypracoval samostatně pod vedením vedoucího semestrálního projektu a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2001 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne .....

podpis autora

Poděkování

Děkuji vedoucímu práce panu doc. Ing. Danu Komosnému, Ph.D. za velmi užitečnou metodickou pomoc a cenné rady při zpracování diplomové práce.

V Brně dne ……

OBSAH

ÚVOD

Ve své diplomové práci jsem se zabýval problematikou určení fyzické polohy stanice v síti Internet. Metody určení fyzické polohy stanice rozdělujeme na aktivní a pasivní. Aktivní metody spočívají v měření zpoždění v síti. Ve své práci jsem se zaměřil na pasivní metody zjišťování polohy stanice, tj. pomocí geolokačních databází. Realizoval jsem také aplikaci, která umí určit fyzickou polohu potenciálního útočníka na operační systém.

V prvních třech kapitolách jsem se věnoval teoretickému rozboru problému lokalizace potenciálního útočníka na operační systém. V první kapitole jsem se zabýval pojmem operační systém, zmínil jsem některé současné operační systémy. U dvou nejběžnějších operačních systémů, u OS Windows a OS Linux, jsem představil jejich obecnou architekturu. V první kapitole také pojednávám o možných způsobech útoku na operační systém: od zasílání spamu a zavirovaných emailů, prohledávání konferencí a mailing listů, přes prohledávání databáze Whois až k hromadným pingům a skenování portů. Hacker se může také pokusit o fyzický přístup k počítači či serveru.

V druhé kapitole pojednávám o programech sloužících k monitorování sítě. Popsal jsem použití síťového skeneru Nmap, což je víceúčelový program, který se používá ke skenování portů a zjišťování, které síťové programy běží na konkrétním počítači. Nmap umožňuje využívat různé druhy skenování portů. Lze jím také určovat typ a verzi operačního systému. Dále jsem v práci popsal detektor skenování, program Scanlogd. Tento program umí detekovat útok a příslušné údaje zapíše do systémových logů. K vyhodnocení údajů jsem využil program Swatch.

Třetí kapitola se zabývá problematikou určení fyzické polohy stanic v Internetu. V kapitole jsou zmíněny aktivní a pasivní metody geolokace. Z aktivních metod geolokace jsem zmínil metodu Geoping a metodu Constraint Based Geolocation (CBG). Druhý typ - pasivní metody geolokace spočívají v zjištění pozice stanice pomocí databází. Geolokační databáze rozdělujeme na ty s volným přístupem a na komerční. Z databází s volným přístupem bych zmínil databázi Whois, která vzniká díky registrátorům domén. Dalšími databázemi jsou například databáze firmy MaxMind. Firma MaxMind nabízí jak komerční verze, tak i verzi volně dostupnou, kterou využívám ve své aplikaci. U databáze MaxMind je diskutována přesnost lokalizace stanice.

Čtvrtá a pátá kapitola jsou praktickým ověřením možností určení fyzické polohy potenciálního útočníka na operační systém. Ve čtvrté kapitole popisuji strukturu vytvořené geolokační aplikace. Aplikace je rozdělena do tří modulů: modul `asimul`, provádějící simulaci útoku z datasetu reálných IP adres, modul `adetector`, sloužící k detekci útoku, a modul webové stránky, zajišťující výstup do mapy. Aplikace je určena pro platformu operačního systému Linux.

Závěrečná kapitola se zabývá výsledky vlastního měření. Díky tomu, že jsme, po dohodě s vedoucím práce, porovnali mezi sebou výsledky ostatních zadání jiných metod, a tím bylo možné porovnat mezi sebou jak pasivní, tak i aktivní metody geolokace.

1. METODY ÚTOKU NA OPERAČNÍ SYSTÉM

Operační systém je velmi komplexní software, který se obecně skládá z programového jádra (kernel) a pomocných programových modulů, systémových nástrojů, jejichž cílem je společně s jádrem vytvořit stabilní aplikační rozhraní (API) pro provádění aplikačních programů zadávaných uživatelem.

Vývoj operačních systémů je těsně svázán s vývojem hardware, na který se musí reagovat. Jako příklad rozdílných přístupů při vývoji hardware je možné uvést Harvardskou architekturu, kde paměť pro data a paměť pro instrukce programu mohou mít odlišné vlastnosti zejména v délce slova a časování a architekturu von Neumannovu, kde je pro data a instrukce programu vyhrazena společná paměť a přístup do paměti je sekvenční. V poslední době se prosazuje z řady důvodů Harvardská architektura, která přináší možnosti rychlého zpracování velkého objemu dat pomocí paralelních procesů na vícejádrových procesorech.

Do této skupiny hardware mezi špičkové výrobky současnosti určené pro všestranné využití patří například šestijádrový procesor Intel Core i7-3970X se základní frekvencí 3,5 GHz, který zpracovává současně až dvanáct programových vláken pomocí instrukčního souboru SSE4 [9].

1.1 Současné operační systémy

V současnosti je v závislosti na hardware provozována celá řada operačních systémů. Mezi hlavní aktuální operační systémy, co se týče počítačů, lze uvést například tyto programové produkty: od firmy Microsoft Windows XP, Windows Vista (70 milionů řádků programového kódu), Windows 7 a nejnověji Windows 8, dále Mac OS X pro počítače Apple, firma IBM dodává i5/OS a otevřený systém AIX 5L, kompatibilní s UNIXem dle normy Single UNIX Specification verze 3, od firmy UnXis UnixWare 7.1.4, od firmy Sun Microsystems Solaris 11. Dále také open source operační systémy: Ubuntu Linux 12.10, Fedora 17, Debian 6.0.6, FreeBSD 9.0 a další.

Pro operační systémy typu Windows je charakteristickým rysem využití vrstvové struktury pro práci systému. Vzhledem k tomu, že současné procesory pro zajištění bezpečnosti podporují dva mody činnosti - omezený pro aplikace a privilegovaný pro jádro, se architektura tohoto operačního systému dělí na dvě základní vrstvy s vymezenými režimy. Vrstva uživatelského modu a vrstva jádra (kernel) se svým specifickým režimem činnosti, často označovaným jako kernel mod. Oddělení obou vrstev a současně jejich spolupráci má pro operační systém Windows na starosti program jádra **NTOSkrnl.exe**, který mimo jiné provádí rozdělování času běhu CPU mezi obě výše uvedené vrstvy.

Na základě této architektury vrstev, aplikační programy nemají žádný přímý přístup k hardware ani k operační paměti. Požadavky aplikačních programů jsou postupovány hlouběji do příslušných částí systému. Řízení procesů probíhá pomocí rozhraní Win32-API a vrstvy NTOS.

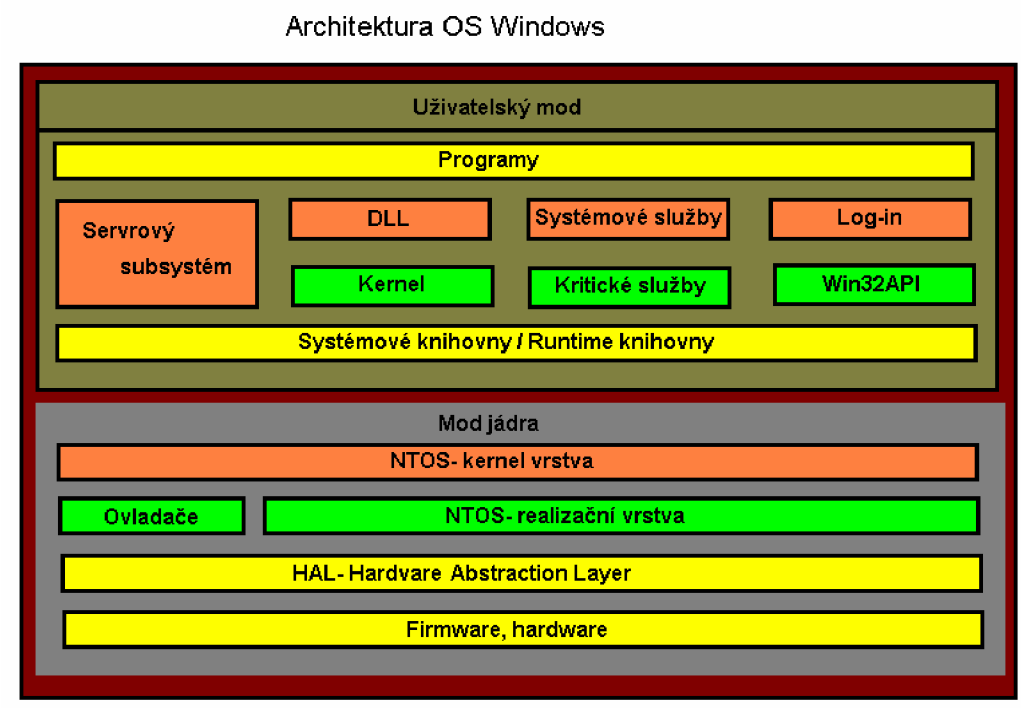
Z důvodu bezpečnosti operačního systému před hackerskými útoky se od programového produktu Vista začínají zavádět pro spouštění procesů v architektuře systému určitá omezení: žádný proces nesmí

být spuštěn s právy administrátora bez potvrzení uživatele a dále byly zavedeny digitální podpisy chráněných procesů.

Windows 7 s sebou přináší další zásadní změnu v architektuře tím, že původní rozsáhlé jádro nahrazuje mikrojádrem s označením MinWin o sto souborech s celkovou velikostí 25MB [14]. Nové mikrojádro vychází z původního jádra pro osvědčené Windows NT a přináší snížení nároků na operační paměť a tím zrychlení celého systému, rovněž i zvýšení schopnosti celého operačního systému udržet se v chodu při vzniku závažné chyby.

Z hlediska architektury operačního systému Windows 8 přinášejí novou funkci *Prostory úložišť:* data mohou být současně ukládána až na tři různé disky, kromě vyměnitelných [12]. Obecnou architekturu systému Windows můžeme vidět na obrázku 1.1.

V privilegovaném módu běží vrstva služeb, výkonná část a jádro. Výkonná část plní základní systémové funkce, nad nimi operují serverové subsystémy, podpora aplikačních rozhraní a podsystémy prostředí. Serverové subsystémy běží v uživatelském módu. Modulová struktura umožňuje měnit podsystémy prostředí bez ovlivnění jádra [3].

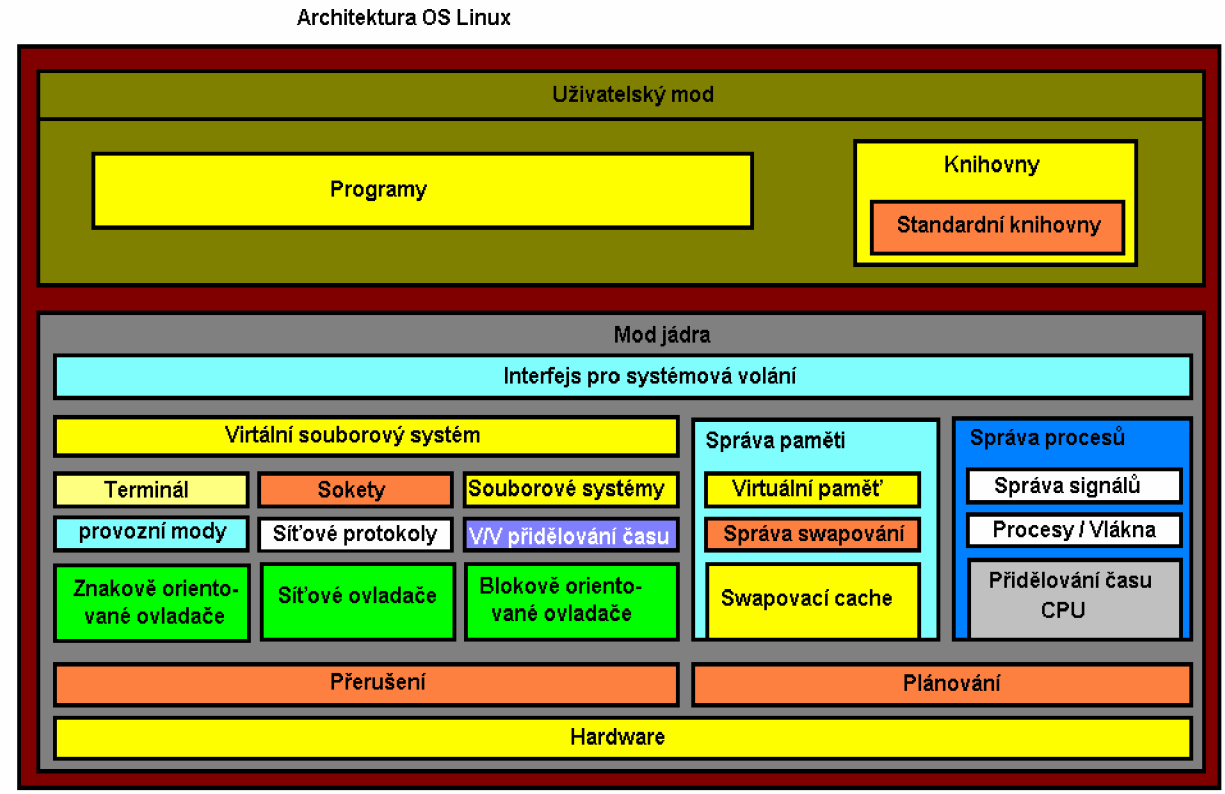


Obr. 1.1: Architektura OS Windows [3].

Operační systém Linux vychází z unixového systému Minix. Jádro Linuxu bylo navrženo finským studentem Linusem Torvaldsem, který v roce 1991 vydal jeho první verzi pod označením 0.01 [10]. Ve své obecné podobě má architekturu zachycenou obrázkem 1.2.

Opět jsou zde dvě základní vrstvy označované v literatuře jako uživatelský mod nebo pojmem uživatelský režim a vrstva modu či režimu jádra, tyto dvě základní vrstvy jsou namapovány na jednotlivé

režimy procesoru. Architektura procesorů x86 umožňuje čtyři režimy označovaní jako *RING0* až *RING3*.



Obr. 1.2: Architektura OS Linux.

1.2 Problematika útoků po síti

Internet přináší dříve netušené možnosti komunikace, získávání a publikování informací, možnosti spolupráce a propojení pomocí sociálních sítí, toto souhrnně označujeme jako Web 2.0. Spolu s výhodami jsou zde i rizika spojená se spamem, podvodnými stránkami, viry a červy, hackery. Právě na možnosti útoků hackery bych se zaměřil v této kapitole.

Původní význam slova hacker byl pro počítačové nadšence, zejména ty, kteří měli možnost pracovat s operačním systémem Unix, v Unix éře. Sám Linus Torvalds, první vývojář linuxového jádra, se sám označuje za hackera. Postupem času byl význam slova posunut směrem k „špatným hochům“ a nyní vnímáme tento význam negativně.

Jednou z možností, jak hacker může „oslovit“ mnoho lidí, je zasílat spam. Spam je zaslán do spousty schránek, a z těch uživatelů, kteří na něj odpoví, jsou potom hackerem vybírány vhodné cíle. Některé zprávy jsou spojeny s podvodnými odkazy. Při jejich prozkoumávání, což se u spamu výslovně nedoporučuje, může dojít k infikaci nějakým virem, instalaci zadních vrátek, apod.

Bude-li obsah cizího počítače pro hackera nějakým způsobem zajímavý, tak si dá práci s prohledáváním různých konferencí a mailing listů. Občas se stává i zkušeným uživatelům, že potřebují s něčím poradit. Ve snaze získat relevantní radu přikládají chybová hlášení, hlášení jádra, soubory, popisují konfiguraci. Hacker pak snadno získá informace, a to často včetně IP adres zařízení.

Na základě získaných informací mohou být pro hackera zajímavé údaje v databázi Whois, zejména zajímá-li se o konkrétní osobu či organizaci. V Whois databázi si zjistí kontakty, jména nameserverů, data vytvoření a poslední modifikace záznamů. Data vytvoření resp. poslední modifikace mohou napovídat o zabezpečení systémů. Systémy, které byly vytvořeny a modifikovány dávno, pravděpodobně používají staré softwarové vybavení, a naopak, systémy čerstvě vytvořené možná ještě nejsou dostatečně zabezpečené.

Celkem jednoduchá je také metoda hromadného pingu, která spočívá v dotazování všech zařízení na síti. Poslouchá-li na adrese nějaký počítač, tak odpoví a je zjištěna platnost adresy. Existují dva typy ping dotazů: ICMP ping a echo port ping. K dispozici je i několik nástrojů urychlující ping, např. Fping a Nmap. Hacker často zjišťuje i cestu k zařízením pomocí programu Traceroute.

Hacker bude chtít zjistit, jaké služby běží na vašem počítači, proto na něj spustí skener portů. Může to být například program Netcat, Strobe, Nmap. Tyto programy mají různé režimy skenování portů. Hacker se pokusí provést detekci operačního systému, k tomu mu poslouží například program Nmap, případně spuštěný protokol SNMP (Simple Network Management Protocol), či program Queso nebo pasivní tester Siphon [6].

Linux podporuje vzdálené volání procedur RPC (Remote Procedure Call), umožňující počítačům přes síť volat procedury na vzdálených strojích. Tyto procedury nemají vyhrazené porty, registrují se pomocí mapovače portů. Hacker se může dotazovat mapovače portů pomocí programu Rpcinfo, případně Nmapem. Na linuxových strojích bývají často souborové systémy sdíleny pomocí NFS (Network File System). Hacker může z NFS zjistit topologii sítě, jména dalších hostitelů, možné programové vybavení, uživatelská jména, verze softwaru. Podaří-li se mu nainstalovat do sdíleného souborového systému trojského koně, pak se snadno může dostat ke klientům, kteří si souborový systém připojují.

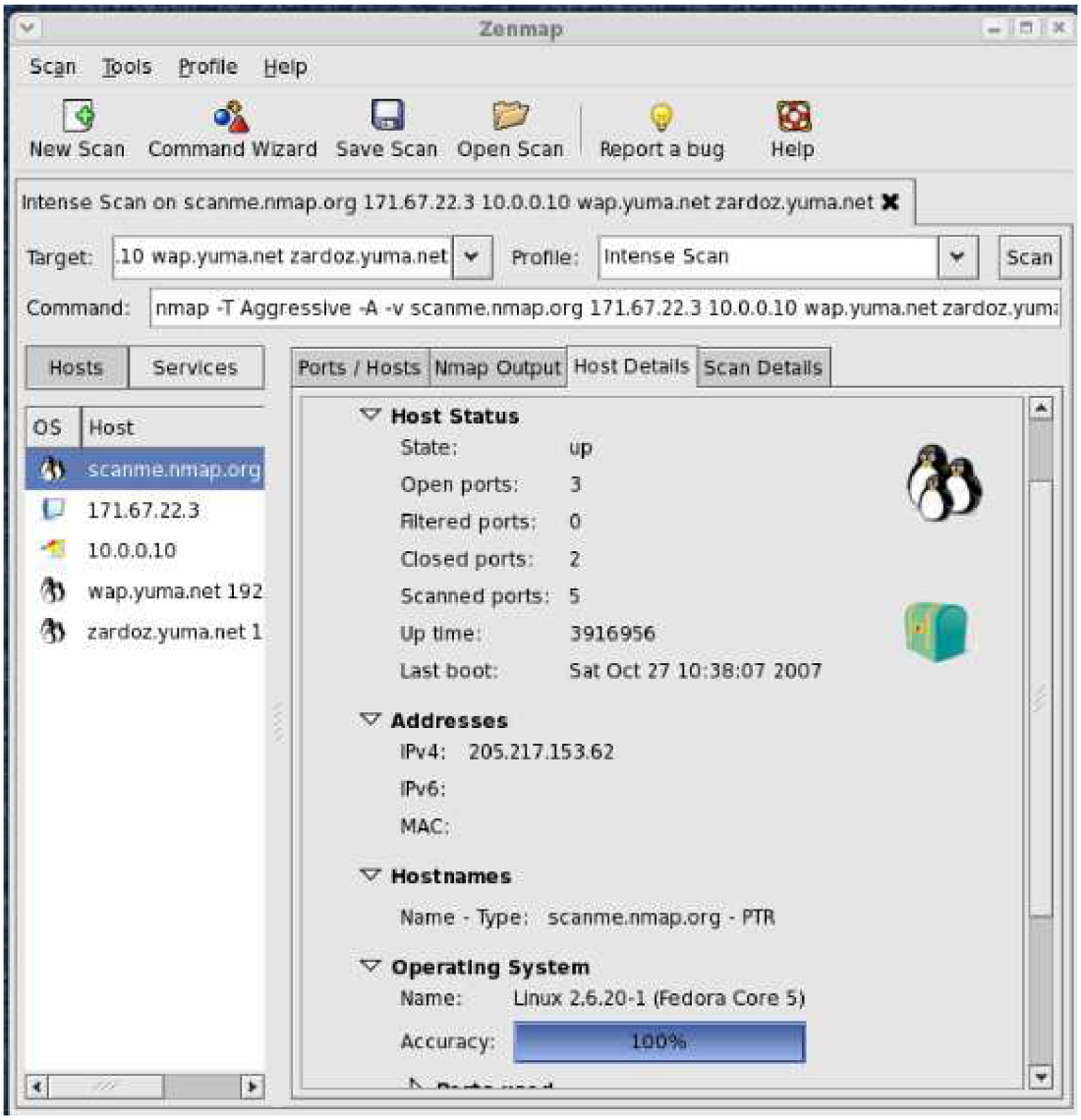
K otestování bezpečnosti počítače slouží síťové bezpečnostní skenery, které se ovšem mohou také stát nástroji hackera. Síťové skenery nezahrnují pouze přímé útoky, ale vše, co by hackerovi mohlo poskytnout užitečné informace, které by mu při útoku mohly pomoci, například uživatelská jména, seznamy nainstalovaného softwaru a běžící programy. V současnosti se jako síťové skenery používají programy, například ISS (Internet Security Scanner), Satan/SAINT, SARA (Security Auditor´s Research Assistant), Nessus.

Hacker může také využívat různé finty označované jako sociální inženýrství. Třeba do firmy zavolá a vydává se za někoho jiného (kolegu, nadřízeného, pracovníka poskytovatele internetu, aj.). Hacker pro svůj úspěch může podniknout i fyzický útok. Jakmile se dostane k počítači, nastartuje svoji kopii OS z DVD a modifikuje stávající operační systém, například zaznamenává a odesílá stisknuté klávesy. Jedním z prvních cílů hackera je získat na napadeném stroji uživatelský účet. Pokud se mu to podaří, pak se snaží různými metodami posilovat svoje pravomoce, až se dostane k pravomoci privilegovaného uživatele `root`. Jakmile se mu toto podaří, pak má neomezenou kontrolu nad strojem [6].

2. PROGRAMY MONITOROVÁNÍ SÍTĚ

2.1 Nmap

Nmap (Network MAPer) je bezpečnostní skener portů vyvinutý americkým počítačovým expertem Gordonem Lyonem z Kalifornie. Nmap patří do skupiny open source programů. Aktuální je verze 6.25. Nmap běží na systémech Linux, Solaris, BSD, Windows a Mac OS X. Program Nmap se používá k zjišťování stavu sítě a jejich uzlů, například k hledání dosažitelných zařízení, detekci otevřených portů, zkoumání běžících služeb a jejich verzí, zjišťování způsobu manipulace s pakety apod. Provozován je v textovém nebo grafickém prostředí Zenmap, viz obr. 2.1.



Obr. 2.1: Grafické prostředí programu Nmap - Zenmap.

2.1.1 Specifikace cíle

Vše, co je na příkazovém řádku Nmapu, pokud to není volba (nebo její argument), je považováno za specifikaci cíle. Nejjednodušší případ pro skenování je udat IP adresu nebo název počítače (hostname).

Pokud chceme proskenovat celou počítačovou síť, Nmap podporuje Classless Inter-Domain Routing (CIDR) adresaci. Můžeme připojit masku /<numbits> k IPv4 adrese nebo k hostname. Nmap proskenuje každou IP adresu, pro kterou prvních n bitů masky <numbits> je stejných jako u referenční IP adresy nebo hostname. Například 192.168.5.0/24 bude skenovat 256 zařízení mezi 192.168.5.0 a 192.168.5.255. Největší hodnota je /32, která udává jedno zařízení. Nejmenší hodnota je /0, která adresuje celý Internet.

CIDR notace je krátká, ale ne vždy dostatečně flexibilní. Například chceme-li projít síť 192.168.0.0/16, ale přeskočit adresy končící na .0 a .255, protože adresa končící 0 udává adresu sítě a adresa končící 255 udává broadcast adresu. Nmap umožňuje tato zadání, díky možnostem volby rozsahů v jednotlivých oktetech IP adresy. Například zadání 192.168.0-255.1-254, přeskočí adresy končící na 0 nebo 255. Jiný příklad 192.168.2-5,8.1 proskenuje následujících pět adres 192.168.2.1; 192.168.3.1; 192.168.4.1; 192.168.5.1 a 192.168.8.1. Jedna nebo druhá strana rozsahu může být vynechána, pak se standardně použije 0 vlevo, případně 255 vpravo.

IPv6 adresy mohou být specifikovány pouze plně kvalifikovanou IPv6 adresou nebo pomocí hostname. CIDR a oktetové rozsahy zatím pro adresy IPv6 nejsou podporovány.

Zatímco cíle bývají obvykle specifikovány na příkazové řádce, následující zadání je rovněž možné:

`-iL <input filename>` (Input from list)

Nmap načte specifikaci cílů ze vstupního souboru <input filename>. Předávání velkého počtu adres zařízení z příkazového řádku je zdlouhavé, je lepší adresy zadat přímo ze souboru. Položky v souboru mohou být v libovolném tvaru akceptovaném Nmapem (IP adresa, hostname, CIDR, IPv6, nebo rozsahy v jednotlivých oktetech). Každá položka v souboru musí být oddělena jednou nebo více mezerami, tabulátory nebo novými řádky. Je možné místo jména souboru zadat pomlčku (-), Nmap bude potom jména cílů načítat ze standardního vstupu. Vstupní soubor může obsahovat komentáře uvozené znakem #.

`-iR <num hosts>` (Choose random targets)

Chceme-li provádět nějaké průzkumy na Internetu, pak můžeme využít náhodnou volbu cíle. Argument <num hosts> říká Nmapu, kolik IP adres má generovat. Nevhodné IP adresy, například z privátních rozsahů, multicast, nebo nepřidělené rozsahy jsou automaticky přeskočeny. Zadáme-li do <num hosts> 0, pak bude probíhat skenování celého Internetu. Je nutno vzít v potaz, že mnozí administrátoři skenování jejich sítí nepřipouštějí a mohou si stěžovat.

`--exclude <host1>[,<host2>[,...]]` (Exclude hosts/networks)

Volba specifikuje čárkou oddělený seznam cílů, které mají být vyjmuty ze skenování. Seznam používá normální syntaxi Nmapu. Tato možnost může být užitečná, když se chci vyhnout systémům a podsítím, které spravuje jiná osoba [11].

`--excludefile <exclude_file>` (Exclude lists from file)

Poskytuje stejnou funkčnost jako `–exclude`, pouze cíle, které mají být přeskočeny, se načítají ze souboru.

2.1.2 Detekce zařízení

Jedním z prvních kroků v průzkumu sítě je zredukovat často značné množství IP adres na seznam aktivních nebo zajímavých zařízení. Skenování každého portu každé adresy je pomalé a není většinou nutné. Samozřejmě, to, co dělá zařízení zajímavé, závisí na účelu skenování. Administrátor se často spokojí s použitím ICMP pingu k určení zařízení na jeho vnitřní síti. Jinak je situace u externího útočníka, který může spouštět množství sond a snažit se obejít omezení daná firewallem.

Jelikož detekce zařízení je velmi různorodá, proto Nmap nabízí široké množství detekčních technik. Detekce zařízení bývá často nazývána ping scan, ovšem techniky jsou mnohem různorodější než použití zprávy ICMP echo, kterou používá příkaz `ping`. Uživatelé mohou zcela přeskočit ping, díky list scanu (`–sL`) nebo zakázáním pingu (`–Pn`), nebo jen skenováním sítě určitými typy protokolů: TCP SYN/ACK, UDP, SCTP INIT a ICMP sond. Cílem těchto sond je vyžádat odezvy, které ukazují, že je daná adresa aktivní. Na mnoha sítích bývá pouze malé procento zařízení aktivní (z celého rozsahu sítě).

Nespecifikujeme-li žádné volby detekce zařízení, pak Nmap posílá požadavek ICMP echo, TCP SYN paket na port 443, TCP ACK paket na port 80 a ICMP timestamp požadavek. Při použití IPv6 není použita ICMP timestamp, protože není součástí specifikace ICMPv6. Toto standardní nastavení dotazu odpovídá volbám `–PE –PS443 –PA80 –PP`. Výjimkou jsou dotazy ARP (IPv4) a dotazy Neighbor Discovery (IPv6), které jsou použity na lokální síti. Pro neprivilegované uživatele Unixu je standardní sonda TCP SYN paket na port 80 a 443 zavolána pomocí systémového volání `connect`. Tento typ detekce zařízení je obvykle dostatečný na lokální sítí, nicméně pro bezpečností audit je vyžadováno obsáhlejší zahrnutí různých sond.

Volby `–P*` (výběr typu pingu) mohou být kombinované. Šanci průchodu firewallem můžeme zvýšit posláním mnoha sond s různými TCP porty/příznaky a ICMP kódy. Nmap standardně skenuje porty všech nalezených online zařízení [11].

`–sL` (List Scan)

Tato volba vypíše seznam detekovaných zařízení na síti, neposílá však žádný paket na cílové zařízení. Jména zařízení Nmap standardně zjišťuje reverzním DNS dotazem. Na konci skenování Nmap vypíše celkový počet nalezených IP adres.

`–sn` (No port scan)

Volba zamezí Nmapu provádět skenování portů. Nmap pouze vypíše seznam aktivních zařízení. Toto je často označováno „ping scan“, nicméně je možné si vyžádat výpis trasy paketu (traceroute) nebo spuštění NSE skriptů Nmapu. Tato volba dovoluje lehký průzkum sítě, kdy na sebe příliš neupozorníme.

Seznam aktivních strojů je pro útočníka často důležitější než soubor oskenovaných portů na těchto strojích. Volba `-sn` je lepší než ping na broadcast adresu, protože spousta zařízení na broadcast dotazy neodpovídá.

`-Pn` (No ping)

Volba zcela přeskočí fázi detekce zařízení. Standardně Nmap provádí složitější testování jako je skenování portů, detekce OS, aj. pouze na zařízeních, která jsou aktivní. Touto volbou ho například na síti třídy B (`/16`) donutím skenovat všech 65 536 IP adres.

`-PS<port list>` (TCP SYN Ping)

Tato volba zašle prázdný TCP paket s nastaveným příznakem SYN. Standardní cílový port je 80 (konfigurovatelné při překladu – položka `DEFAULT_TCP_PROBE_PORT_SPEC` v `nmap.h`). Jiné porty mohou být specifikovány pomocí parametru. Mezi volbou `-PS` a seznamem portů se nedělá mezera. Je-li specifikováno více sond, budou poslány nezávisle na sobě. Příznak SYN naznačuje vzdálenému systému, že se pokoušíme o spojení. Běžně bude cílový port uzavřen, potom bude zpět zaslán paket s příznakem RST. Je-li port otevřený, cíl přejde do druhé fáze otevírání spojení a zašle TCP paket s příznaky SYN/ACK. Stroj, kde je spuštěn Nmap, poté přeruší vznikající spojení zasláním TCP paketu s příznakem RST. TCP paket s příznakem RST je zaslán jádrem stroje, kde je spuštěn Nmap, jako odezva na neočekávaný paket SYN/ACK. Nmap se v tomto případě nestará o to, zda-li je port otevřený či zavřený. Buď RST nebo SYN/ACK odezva potvrdí Nmapu, že zařízení je dostupné a reaguje.

V systémech Unix může pouze privilegovaný uživatel `root` posílat a přijímat přímo TCP pakety. Pro neprivilegované uživatele se použije systémové volání `connect`. Jestliže `connect` vrátí úspěch nebo chybu `ECONNREFUSED`, pak transportní vrstva obdržela SYN/ACK, respektive RST, tedy zařízení je dostupné. Dojde-li při pokusu o navázání spojení k vypršení času, pak je cíl označen jako neaktivní.

`-PA <port list>` (TCP ACK Ping)

TCP ACK ping je podobný předchozímu TCP SYN pingu. Rozdíl je v tom, že v TCP paketu je místo SYN příznaku nastaven příznak ACK. TCP paket s příznakem ACK se tváří, že potvrzuje data na existujícím TCP spojení, přitom ale žádné neexistuje. Vzdálený stroj by tedy měl vždy odpovědět RST paketem. Volba `-PA` používá standardně stejný port jako sonda SYN (port 80), lze ji také stejně zadat seznam portů k otestování.

Důvod, proč Nmap nabízí oba typy: sondu SYN a sondu ACK, je k obejití firewallu. Mnoho běžných správců konfiguruje firewally, aby nepropouštěly TCP pakety s příznakem SYN. Tímto zamezí běžným útokům na své zařízení, přičemž uživatelé zevnitř sítě nejsou v přístupu na Internet nijak omezováni. Tento nestavový přístup je výpočetně snadný, tedy je implementován v řadě hardwarových i softwarových firewallech. Linuxový Netfilter/iptables software nabízí volbu `--syn` k zablokování těchto paketů. Když je tedy nasazen nestavový firewall (paketový filtr), pak sonda SYN neprojde,

zatímco sonda ACK ano. Naopak u stavového firewallu (na transportní vrstvě) projde sonda SYN, zatímco sonda ACK neprojde. Řešením tohoto dilema je posílat jak sondy SYN, tak sondy ACK.

`-PU <port list>` (UDP Ping)

Volba provede UDP ping, což posílá UDP pakety na zadané porty. Pro většinu portů je paket prázdný, zatímco pro několik portů je v něm specifická výplň. Volba `--data-length` může být použita k zasílání náhodné výplně o zadané délce, nebo pokud specifikuji 0, tak se zakážou výplně úplně. Seznam portů je stejný jako u předchozích probíraných voleb `-PS` a `-PA`. Není-li žádný port specifikován, pak se standardně použije port 40 125. Tento standardní port může být při překladu nastaven změnou `DEFAULT_UDP_PROBE_PORT_SPEC` v `nmap.h`. Tento neobvyklý port je zvolen z toho důvodu, že otevřený port není vhodný pro tento typ sondy. Hlavní výhodou této sondy, že projde přes firewally, které sledují jen protokol TCP.

`-PY <port list>` (SCTP Init Ping)

Tato volba pošle SCTP paket obsahující minimální INIT blok. Standardní cíl je port 80. Jiné porty mohou být specifikovány parametrem. Pomocí bloku INIT signalizuji vzdálenému systému, že chci navázat spojení. Běžně bude cílový port uzavřený a blok ABORT bude zaslán zpět. Bude-li port otevřený, potom cíl přejde na druhou fázi čtyřcestného handshake a odpoví blokem INIT-ACK. Bude-li mít stroj s Nmapem funkční SCTP zásobník, pak ukončí vznikající spojení zasláním bloku ABORT. Blok ABORT je zaslán jádrem jako reakce na nečekaný blok INIT-ACK.

Nmap nezjišťuje, zda-li je port otevřený či uzavřený. Buď odezva ABORT nebo odezva INIT-ACK říká Nmapu, že zařízení je aktivní a odpovídá. V systémech Unix pouze privilegovaný uživatel `root` je oprávněn přímo manipulovat s pakety SCTP. Použití SCTP INIT Ping není dostupné pro neprivilegované uživatele.

`-PE; -PP; -PM` (ICMP Ping Types)

Nmap posílá paket ICMP typ 8 (echo request) na cílovou adresu a očekává odpověď ICMP typ 0 (echo reply). Naneštěstí pro průzkumníky sítě, mnohá zařízení a firewally blokují tyto pakety, místo aby na ně odpověděly. Tedy pouze ICMP průzkum neznámých cílů na Internetu je obvykle nedostatečný. Nicméně pro systémové administrátory na místní síti může být použití volby `-PE` výhodné.

Dotaz na timestamp a dotaz na masku adresy mohou být zaslány pomocí volby `-PP` resp. `-PM`. Odpověď timestamp (ICMP kód 14) nebo odpověď na masku adresy (kód 18) prozradí, že zařízení je dostupné. Tyto dva dotazy mohou být výhodné, když administrátor zablokuje odezvu na echo, ale zapomene na tyto dotazy, které mohou být využity ke stejnému účelu [11].

`-PO <protocol list>` (IP Protokol Ping)

Jedna z dalších metod detekce zařízení je IP protokol ping. Nmap zasílá IP pakety s nastaveným číslem protokolu v IP hlavičce. Seznam protokolů má stejný formát jako seznam portů dříve zmíněných. Není-li žádný protokol uveden, pak je standardně posláno několik IP paketů pro ICMP (protokol 1),

IGMP (protokol 2) a IP v IP (protokol 4). Pro protokoly ICMP, IGMP, TCP (protokol 6), UDP (protokol 17) a SCTP (protokol 132) jsou pakety poslány včetně příslušných protokolových hlaviček, zatímco pro ostatní protokoly je zaslána jen IP hlavička (pokud není specifikována volba `--data-length`).

Tento způsob testování očekává buď odezvu shodným protokolem anebo ICMP protocol unreachable hlášky, které signalizují, že daný protokol na cílovém zařízení není implementován. Jeden nebo druhý typ odezvy signalizují, že cíl je aktivní.

`-PR` (ARP Ping)

Jedno z nejčastějších použití Nmapu je skenování ethernet LAN. Na většině LAN, zvláště na těch, které používají privátní rozsahy je většina adres nevyužitá. Když Nmap zkusí poslat IP paket jako například ICMP echo request, operační systém musí zjistit cílovou hardwarovou (ARP) adresu odpovídající hledané IP adrese, tak aby mohl být spolehlivě doručen ethernetový rámec. Tento způsob je obvykle pomalý a problematický, protože operační systémy nebyly napsány k řešení spousty ARP dotazů na neexistující zařízení, a to v krátkém čase.

Nmap provádí ARP dotazy sám, optimalizovanými algoritmy. Pokud dostane odezvu, potom už nemusí provádět IP ping, protože ví, že zařízení je aktivní. ARP sken je mnohonásobně rychlejší a spolehlivější než skeny založené na IP protokolu. Nmap používá ARP vždy, pokud skenuje lokální síť. Chceme-li toto zakázat, je nutno použít volbu `--disable-arp-ping`. Pro IPv6 používá Nmap Neighbor Discovery.

`--traceroute` (Trace Path to Host)

Zjišťování trasy je prováděno po skenování, s využitím informací ze skenování ke stanovení portu a protokolu, který nejpravděpodobněji dojde k cíli. Pracuje se všemi typy skenů, s výjimkou spojovacích skenů (`-sT`) a nečinných skenů (`-sI`). Zjišťování cesty probíhá paralelně.

K zjištění cesty posílá pakety s nízkým TTL (Time to Live) jako pokus vyvolat ICMP Time Exceeded zprávy z mezilehlých skoků mezi skenerem a cílem. Standardně se při zjišťování cesty začíná s TTL rovno jedné a zvyšuje se o jednotku, než se dosáhne cíle. Nmap při zjišťování cesty začíná s vysokým TTL a postupně ho snižuje k nule, což mu umožňuje využít optimalizované algoritmy k urychlení hledání cesty. V průměru Nmap pošle o 5 až 10 paketů na jedno zařízení méně než ostatní programy na zjišťování cesty. Vše ovšem záleží na stavu sítě.

`-n` (No DNS Resolution)

Volba zakáže Nmapu provádět reverzní DNS vyhledávání aktivních IP adres. Jelikož DNS může být pomalé, tato možnost může radikálně snížit čas skenování [11].

`-R` (DNS Resolution for All Targets)

Volba přikáže Nmapu provádět DNS vyhledávání na všechny cílové adresy. Standardně se reverzní překlad DNS provádí jen na aktivních zařízeních.

`--system-dns` (Use System DNS Resolver)

Standardně, Nmap zjišťuje názvy zasíláním dotazů přímo našim nakonfigurovaným DNS serverům. Mnoho dotazů je posíláno paralelně, aby se zvýšil výkon. Zvolíme-li tuto možnost, pak bude využit náš systémový resolver. Toto je pomalejší a zřídkakdy užitečné. Náš systémový resolver je použit vždy, když děláme IPv6 sken.

`--dns-servers <server1>[,<server2>[,...]]` (Servers to Use for Reverse DNS Queries)

Standardně si Nmap zjistí naše DNS servery ze souboru resolv.conf (Unix) nebo z Registru (Windows). Alternativně můžeme použít tuto volbu a specifikovat alternativní servery. Tato volba není respektována, pokud použijete `--system-dns` nebo IPv6 sken [11].

2.1.3 Stavy portů

Cílem programu Nmap je být efektivním skenerem portů. Jednoduchý příkaz `nmap <target>` skenuje 1 000 TCP portů na zařízení `<target>`. Zatímco ostatní skenery většinou rozlišovaly porty na otevřené nebo zavřené, Nmap nabízí podrobnější rozlišení. Nmap rozlišuje šest stavů: `open`, `closed`, `filtered`, `unfiltered`, `open|filtered`, nebo `closed|filtered`.

Tyto stavy nejsou vlastností samotných portů, pouze popisují, jak Nmap tyto porty vidí. Například Nmap sken ze stejné sítě vidí port `135/tcp` jako `open`, zatímco sken z Internetu ho vidí jako `filtered`.

Stav portu `open`:

Udává, že aplikace aktivně přijímá TCP spojení, UDP datagramy nebo SCTP asociace na tomto portu. Tyto nálezy jsou primárním cílem skenování portů. Lidé zabývající se bezpečností vědí, že každý otevřený port představuje bránu pro útok. Útočníci se snaží zneužít otevřených portů, zatímco administrátoři se je snaží buď uzavřít nebo prostřednictvím firewallu ochránit způsobem, který neomezí legitimní uživatele. Otevřené porty jsou důležité i z pohledu obyčejného uživatele, protože ukazují služby, které se v dané síti dají využít [11].

Stav portu `closed`:

Uzavřený port je dostupný (obdrží a odpoví na Nmap sondy), ale není zde žádná aplikace naslouchající na tomto portu. Uzavřené porty mohou být užitečné tím, že potvrdí, že zařízení na dané IP adrese je aktivní (detekce zařízení, ping sken). Porty také mohou posloužit k detekci operačního systému. Uzavřené porty se vyplatí za nějakou dobu opět proskenovat, protože existuje možnost, že na nich bude spuštěna nějaká aplikace v budoucnu. Administrátoři se mohou rozhodnout takové porty blokovat firewallem, potom je Nmap zařadí do kategorie `filtered`.

Stav portu `filtered:`

Nmap nemůže určit, zda-li je port otevřený, protože filtrování paketů zabrání sondám v dosažení portu. Tyto porty frustrují útočníka, protože podávají tak málo informací. Někdy odpoví ICMP chybovým hlášením (jako typ 3, kód 13 – destination unreachable). Filtry, které jednoduše zahodí paket, jsou také časté. Toto nutí Nmap vícekrát opakovat sondy, aby vyloučil chyby vzniklé přetížením sítě. Tato volba dramaticky zpomaluje sken.

Stav portu `unfiltered:`

Tento stav znamená, že port je dostupný, ale Nmap není schopen určit, zda je port uzavřený nebo otevřený. Pouze ACK sken, který projde firewallem, klasifikuje porty na tento typ. Následné skenování unfiltered portů jinými sondami může pomoci rozhodnout, zda je port otevřený.

Stav portu `open|filtered:`

Nmap přiřadí tento stav, pokud není schopen určit, zda je port otevřený nebo filtrovaný. Toto se stává u skenů, kdy otevřený port nedává žádnou odezvu. Nepřítomnost odezvy může také znamenat, že paketový filtr zahodil sondu, případně odezvu, kterou vyvolala. UDP, IP sondy, FIN, NULL a Xmas skeny klasifikují porty tímto způsobem.

Stav portu `closed|filtered:`

Tento stav Nmap vypíše, pokud není schopen rozhodnout, zda-li je port uzavřený nebo filtrovaný [11].

2.1.4 Specifikace portů a skenování

Nmap nabízí volby pro specifikaci, které porty skenovat, zda-li skenovat náhodně nebo sekvenčně. Nmap standardně skenuje 1 000 nejběžnějších portů pro každý protokol.

`-p <port ranges>` (Only Scan Specified Ports)

Tato volba specifikuje, které porty chceme skenovat, a potlačuje standardní chování. Je možné zadat porty individuálně, nebo jako rozsah (např. 1-1023). Počáteční či koncová hodnota rozsahu může být vynechána, Nmap potom použije 1 resp. 65 535. Skenování portu 0 je možné, pokud ho explicitně specifikujeme. Pro skenování IP protokolem (volba `-sO`) je možné specifikovat čísla protokolů (0-255), které chceme použít.

Skenujeme-li více protokolů (např. TCP a UDP), musíme specifikovat předponu protokolu před příslušné porty: `T:` pro TCP, `U:` pro UDP, `S:` pro SCTP, `P:` pro IP protokol. Nezadáme-li žádnou předponu protokolu, pak budou příslušné porty přidány pro všechny protokoly. Porty mohou být specifikovány také pomocí jména, které je uvedeno v `nmap-services`. Se jmény se dají použít i divoké znaky ? a *. Například chceme proskenovat FTP porty a všechny porty, které začínají na http. Potom zadám volbu `-p ftp,http*`. Rozsahy portů mohou být také označeny hranatými závorkami, tyto rozsahy potom definují porty uvedené v souboru `nmap-services`.

`-F` (Fast Scan)

Volba ke snížení počtu skenovaných portů, čímž se zrychlí proces skenování. Nmap normálně skenuje 1 000 portů, s touto volbou je počet zredukován na 100. Nmap nahlíží do souboru `nmap-services`, kde si přečte porty s nejčastější frekvencí použití.

`-r` (Don´t Randomize Ports)

Standardně Nmap přistupuje k portům v náhodném pořadí. Tato volba ho přepne na sekvenční přístup (od nejnižších portů směrem k nejvyšším).

`--port-ratio <ratio - decimal number between 0 and 1>`

Proskenuje porty s poměrem větším než `<ratio>`, které musí být mezi 0 a 1.

`--top-ports <n>`

Skenuje `<n>` portů s nejvyšším poměrem `<ratio>`, které jsou uvedeny v souboru `nmap-services`. `<n>` musí být 1 nebo větší [11].

2.2 Scanlogd

Program pochází z dílny Solar Designer. Scanlogd je samostatný skenování detekující démon. K hlídání příchozích spojení může použít schránky (raw sockets) nebo funkce z libnids a libcap.

Scanlogd vyhodnotí jako neoprávněnou obhlídku portů toto: zjištění v případě sedmi privilegovaných portů (<1024), zjištění v případě jedenadvaceti neprivilegovaných portů (>1024) či odpovídající kombinace obojího, to vše v rámci třívteřinových intervalů. Neprodleně zaznamená skenování do souboru systémového logu operačního systému. Pokud objeví více než pět případů skenování za dvacet vteřin, Scanlogd přestane o skenování podávat zprávy v zájmu předejití potenciálního útoku přetížením, který by mohl přeplnit vaše záznamy.

Zápisy v syslogu mají následující formu:

`source_addr to dest_addr ports port, port, ..., TCP_flags @time`

Zmíněné TCP příznaky jsou TCP řídící bity, které jsou nastaveny v samotných paketech. Záznamy mohou být užitečné v případě hlubšího zaobírání se skenováním a útoky [6].

2.3 Swatch

Autorem programu Swatch (the Simple Watchdog) je Todd Atkins. Program Swatch analyzuje protokolové soubory, které pročítá buďto najednou nebo čte pouze naposledy připojované řádky a dokáže také číst výstup z libovolných příkazů.

Swatch je napsaný v Perlu. Konfigurace Swatch je vystavěna ze vzorů a skupin akcí. Vzor musí být platný perlovský regulární výraz, který je kompatibilní se standardními (grep) regulárními výrazy, ale

je robustnější. Kdykoliv řádek odpovídá vzoru, provede se odpovídající akce nebo posloupnost akcí. Několik akcí je uvedeno zde:

`echo`

Vypíše řádek na standardní výstup. Můžeme zvolit i barvu, což je užitečné pro zvýrazňování různých úrovní důležitosti.

`bell`

Terminál, na kterém běží Swatch, pípne.

`mail`

Elektronickou poštou zašle odpovídající řádky jednomu či více uživatelům.

`write`

Vypíše řádky na terminál jednomu či více uživatelům.

`pipe`

Pošle odpovídající řádky na standardní vstup nějakému programu.

`throttle`

Pseudo-akce umožňující nastavit, jak často se musí nějaká položka objevovat, aby byla ukázána více než jednou.

S tímto nastavením bychom mohli spouštět Swatch různými způsoby:

```
swatch –examine=/var/log/syslog
```

Provede jediný průchod protokolovým souborem.

```
swatch –tail-file=/var/log/syslog
```

Zpracuje celý soubor a dále pokračuje na nově připojených položkách.

```
swatch –read-pipe=/tmp/debug_sshd
```

Swatch bude odchytávat a rozebírat výstup programu `debug_sshd`, který jednoduše spouští `sshd` v nevětvícím se debug módu [6].

3. FYZICKÁ POLOHA STANIC V INTERNETU

Propojení počítačů prostřednictvím počítačových sítí začínalo nejprve pomocí místních sítí LAN. V takovéto síti jsou počítače vzdáleny řádově stovky metrů, takže určovat jejich polohu nemá prakticky smysl, poloha všech počítačů je z hlediska zeměpisných souřadnic přibližně stejná. S nástupem globální sítě – Internetu – došlo na potřebu určit zeměpisnou polohu stanice, tj. geolokovat stanici. Důvody pro geolokaci jsou různé, například:

- Online reklama: Reklama je v moderní společnosti velice častým jevem, je součástí všech médií, ať papírových či elektronických. Náklady na reklamu a tím i účinnost reklamy se zvyšuje s rostoucím počtem oslovených občanů. Internet, který se rozvíjí a v dnešní době je moderním trendem, je ideálním místem pro reklamu. Drtivá většina internetových služeb, které jsou nabízeny uživatelům zdarma, je právě placena z reklamy. Jako příklad vezměme vyhledávač Google, který zobrazuje při vyhledávání několik reklamních odkazů. Tyto odkazy se samozřejmě snaží zaměřit na správnou skupinu uživatelů. V tomto mohou pomoci nástroje, jakými jsou vlastní význam zadaného dotazu, analýza předchozích dotazů (u přihlášených uživatelů) a právě také fyzická poloha uživatele.
- Aktuálnost z hlediska polohy, dopravní spojení: Uživatelé, kteří často cestují, ocení aktuální informace, například počasí, dopravní informace, kulturní akce aj.
- VoIP telefonie: IP telefonie a mobilní telefony ukončily klasické rozdělení telefonního volání na místní síť, město a meziměsto. Pomocí čísla pevné linky a telefonního seznamu se dalo určit bydliště volaného. Určení polohy účastníka na telefonu může v krizové situaci zachránit i život. Síť VoIP není ale primárně určena k nouzovým voláním.
- Boj proti spamu: Určení lokality odkud spam přichází.
- Zábava: Na myšlence geolokace vznikají i nové sociální sítě, například Foursquare. Pomocí sítě Foursquare uživatel přijde do nějaké budovy, sociální síť pomoci geolokace zjistí, kteří kamarádi jsou v budově a co dělají, takže se dotyčný může s nimi naživo sejít [17].

3.1 Aktivní metody geolokace

Geolokační techniky rozdělujeme na pasivní a aktivní. Aktivní metody spočívají v měření zpoždění mezi odeslaným paketem ze zdrojové adresy a přijatým paketem v cíli. Změřené zpoždění teoreticky odpovídá vzdálenosti mezi stanicemi. Převod zpoždění na vzdálenost není z praktického pohledu úplně přesný, protože se zde projevují vlivy popsané níže.

Měřit zpoždění, tj. dobu od odeslání k přijetí paketu, je poměrně problematické. Pro přesné změření zpoždění by bylo nutné obě stanice přesně časově synchronizovat. Nebudou-li stanice časově přesně synchronizovány, potom se zpoždění bude jevit větší či menší. Tedy vzdálenost se bude jevit delší či kratší, mohlo by dojít díky chybnému časovému údaji dokonce i k překonání rychlosti šíření v daném médiu. Tuto problematiku se částečně podařilo obejít měřením oboucestného zpoždění, tedy zpoždění

tam i zpět, tzv. round trip time (RTT). Použitím RTT se podařilo vyřešit problém časové synchronizace zdroje a cíle, ovšem použití RTT přináší zase jiné problémy.

Zpoždění se skládá z několika částí. Zahrnujeme do něj zpoždění vznikající při zpracování údajů ve zdroji, čekání ve frontě zdroje, obdobně to platí pro cílovou stanici. Dále je to zpoždění na mezilehlých zařízeních (přepínače, směrovače) a zpoždění dané konečnou rychlostí šíření signálu přenosovým médiem. Balej a Komosný udávají zpoždění na mezilehlých zařízeních takto: na nezatíženém přepínači 1-10 μs a na směrovači 10-100 μs, přičemž na hodně zatíženém prvku mohou být tato zpoždění až několikanásobná [1]. Pro zpoždění způsobené dobou šíření v médiu za předpokladu použití optického vlákna udávají tito autoři rychlost šíření 0,65 násobek rychlosti světla c. Hlavní částí zpoždění, zejména pro velké vzdálenosti, je zpoždění způsobené dobou šíření v médiu. Vzhledem k tomu, že telekomunikační vedení bývají pokládána podél silnic, železnic, dálkových rozvodů elektřiny, je délka těchto telekomunikačních vedení delší než přímá vzdálenost mezi stanicemi. Autoři Balej, Komosný uvádí pro akademickou síť CESNET2 délku vedení přibližně dvojnásobnou oproti skutečné vzdálenosti, [1].

Problém při měření zpoždění RTT nastává díky různorodosti internetu, kdy například může paket vyslaný od cíle ke zdroji putovat jinou cestou a přes jiné mezilehlé prvky než paket vyslaný ze zdroje k cíli. RTT se potom pro zjištění jednocestného zpoždění (a tedy odpovídající vzdálenosti mezi stanicemi) totiž rozdělí jednoduše na poloviny, bez ohledu na jinou dobu šíření po cestě zpět. Zpoždění můžeme také rozdělit na část deterministikou a stochastickou. Část deterministická odpovídá době šíření, zatímco část stochastická je způsobena především různým časově proměnným zatížení prvků v Internetu. Část stochastickou se pokoušíme eliminovat opakovaným měřením a volbou nejmenší hodnoty z naměřených zpoždění. Dovoluje-li to charakter experimentu, je vhodné měřit v různé hodiny v různých dnech v týdnu.

3.1.1 Geoping

Jednou z metod využívající změřené zpoždění ke geolokaci je tzv. Geoping. V metodě se využívá změřené zpoždění, přesněji změřený vektor zpoždění k referenčním bodům.

Pro použití metody Geoping je nutné vytvořit množinu internetových stanic (serverů), tzv. dataset. V tomto datasetu rozlišujeme dva typy stanic – referenční body (landmarky) a sondy (probes). U referenčních bodů potřebuji znát přesně jejich polohu, zatímco u sond stačí orientační informace o poloze. Referenčních bodů je potřeba řádově 100-200, počet sond je volitelný, obvykle 7-10. Autoři Padmanabhan a Subramanian používali 265 „dobře připojených“ referenčních stanic a 2 až 14 sond. Postupným měřením zjistili, že optimum je 7-9 sond - vhodně rozmístěných [15].

Princip metody spočívá v změření zpoždění od sond k referenčním bodům a od cíle k referenčním bodům, výsledkem jsou tzv. vektory zpoždění. Po změření všech vektorů zpoždění pak porovnávám tyto vektory na principu minimálního rozdílu (Eukleidovská vzdálenost). Po nalezení referenčního bodu s vektorem zpoždění nejvíce podobným vektoru zpoždění cíle, prohlásím za pozici cíle pozici tohoto

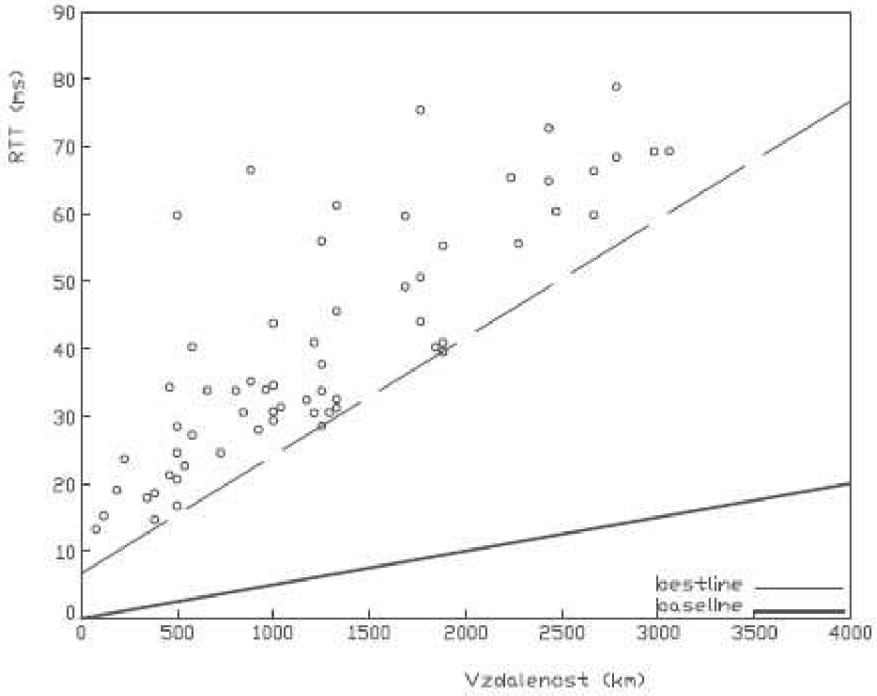
referenčního bodu [15]. V akademickém prostředí se jeví jako výhodné použít jako zdroj referenčních bodů a sond celosvětově budovanou experimentální síť PlanetLab.

3.1.2 Constraint-Based Geolocation

Fyzická pozice může být určena pomocí řady vzdáleností nebo řady úhlů k lokacím, jejichž přesnou polohu znám. Běžným způsobem určování polohy je triangulace, kdy pomocí úhlů a tří referenčních bodů určuji pozici. Jedná-li se vzdálenosti k více bodům, pak mluvíme o multilateraci [5].

Hlavním problémem při multilateraci je přesné změření vzdáleností mezi cílem, který lokalizujeme, a referenčními body. Například systém GPS používá multilateraci ke třem satelitům k určení pozice GPS přijímače. Vzdálenost se určuje pomocí měření doby šíření signálu od GPS satelitu k GPS přijímači. Přesné měření času a časového intervalu je jádrem určování polohy pomocí GPS. V porovnání s GPS je u metody multilaterace problematické přiřazení přesné vzdálenosti vzhledem ke zpoždění v Internetu. Uvažujme množinu referenčních bodů, u kterých známe přesnou pozici. Pro úspěšnou multilateraci musíme převést naměřená zpoždění na vzdálenosti. Zpoždění v Internetu můžeme rozdělit na deterministickou část a stochastickou část. Deterministická část je způsobena samotným šířením signálu v médiu a zpracováním ve směrovačích. Stochastická část je závislá na aktuálním zatížení internetu a spočívá především v čekání ve vstupních frontách síťových zařízení. Nás zajímá část deterministická, kterou lze přibližně zjistit několikanásobným opakováním měření.

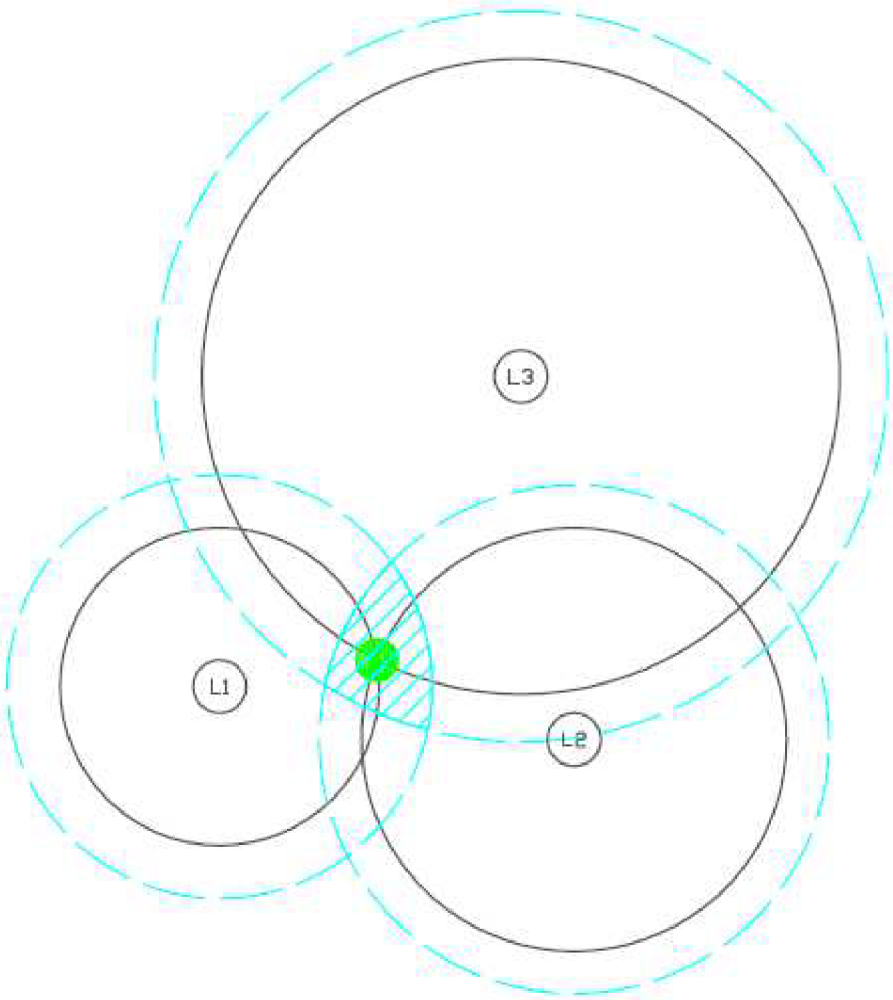
Metoda CBG využívá změřené zpoždění k autokalibraci. Zavádí pojmy baseline a bestline. Baseline odpovídá v grafu zpoždění ideálnímu zpoždění způsobenému pouze šířením v přenosovém médiu, což u optického kabelu odpovídá 2/3 rychlosti světla. Přímka bestline odpovídá nejlepší situaci, tedy nejrychlejšímu šíření paketu z provedených měření Přímka je sestrojena tak, že v oblasti pod přímkou se nesmí vyskytovat žádné změřené zpoždění, viz obr. 3.1. Metoda CBG určí bestline pro každý referenční bod. Toto je princip autokalibrace metody CBG [5].



Obr. 3.1: Autokalibrace metody CBG.

Potom, co proběhla autokalibrace metody CBG, můžeme přejít k vlastnímu určování polohy cílové stanice. Na základě změřeného zpoždění a bestline je určena u každého referenčního bodu vzdálenost od cíle. Sestrojím tedy kružnice od všech referenčních bodů a v jejich průniku se nachází hledaný cíl, viz obr 3.2. V praxi se oblast průniku kružnic převádí na n-úhelník a hledá se jeho těžiště.

Dojde-li k určení oblasti cíle, v tomto případě hovoříme o nadhodnocení oblastí. Může ovšem také dojít k podhodnocení oblastí, kdy se kružnice neprotnou. Nebo k tzv. mismatch, kdy se kružnice protnou, ale cíl leží jinde. Těmto negativním stavům by ale autokalibrace měla zabránit [5].



Obr. 3.2: Určení cíle metodou CBG.

3.2 Pasivní metody geolokace

Pasivní metody geolokace spočívají ve vytvoření databází se záznamy o IP adresách respektive o blocích IP adres. Do těchto databází se potom dotazujeme a zjišťujeme informace o námi hledané konkrétní IP adrese. Databáze rozdělujeme například podle přístupu na volně přístupné a komerční. Počítač v Internetu je charakterizován svojí jedinečnou IP adresou. IP adresy jsou v současné době dvojího druhu IPv4 a IPv6.

V své práci se zabývám IP verze 4. IPv4 adresa je 32 bitové číslo, které udává adresu počítače a adresu sítě. Adresa sítě je dána vlastní IP adresou a maskou sítě. Adresu sítě zjistíme, provedeme-li logický součin (AND) IP adresy a masky sítě. Internet z pohledu těchto adres můžeme rozdělit do těchto tříd:

Tab. 3.1: Třídy IP adres

Třída	**1.bajt**	**Maska**
A	0-127	255.0.0.0
B	128-191	255.255.0.0
C	192-223	255.255.255.0
D	224-239	255.255.255.255
E	240-255	volitelná

Přidělování IP adres je centrálně řízeno organizací Internet Corporation for Assigned Names and Numbers (ICANN). Tato organizace převzala agendu organizace Internet Assigned Numbers Authority (IANA) a dalších agentur. Organizace ICANN se stará o přidělování IP adres, čísla protokolů, generické a národní domény a kořenové servery [18]. ICANN deleguje část svých pravomocí na podřízené organizace. O přidělování IP adres na regionální úrovní se starají Regional Internet Registries (RIRs). RIRové potom přidělují po blocích IP adresy jednotlivým poskytovatelům (ISP) a ti potom koncovým zákazníkům.

3.2.1 Databáze Whois

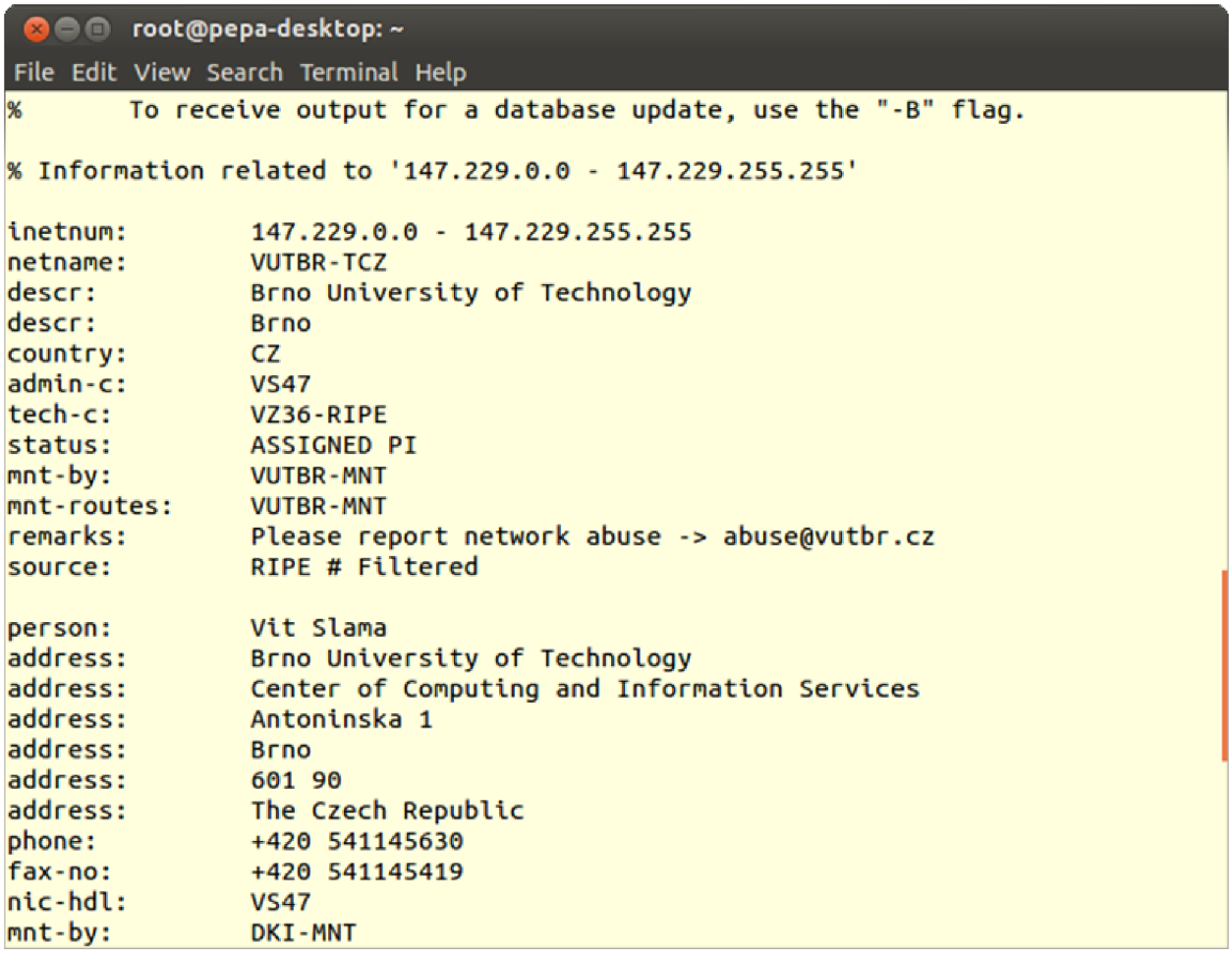
Databáze Whois je veřejně přístupnou databází, která slouží k evidenci údajů o majitelích internetových domén a IP adres. Přístup do ní je zdarma. V databázi jsou uvedeny následující údaje organizace, adresa, kontaktní osoby. Databáze nemá standardizovaný formát položek, například adresa je uvozena jednou pomocí address (pro název, pro město, i pro zemi), u jiných adres je zde uvození příslušnými prvky adresy. S databází se tedy špatně pracuje automatizovanými prostředky. Do databáze přispívají následující regionální internetoví registrátoři (RIR):

- African Network Information Centre (AfriNIC) – pokrývá Afriku,
- American Registry for Internet Numbers (ARIN) – pokrývá Spojené státy, Kanadu, část Karibské oblasti a Antarktidu,
- Asia-Pacifik Network Information Centre (APNIC) – pokrývá Asii, Austrálii, Nový Zéland a přilehlé země,
- Latin America and Carribean Network Information Centre (LACNIC) – pokrývá Latinskou Ameriku a část Karibské oblasti,
- Réseaux IP Européens Network Coordination Centre (RIPE NCC) – pokrývá Evropu, Rusko, Střední Východ a Střední Asii [7].

Výpis z Whois databáze je na obrázku 3.3.

ICANN se stará o aktuálnost dat v databázi Whois implementací Whois Data Reminder Policy (WDRP). ICANN průběžně vyzývá registrátory k tomu, aby přiměli jejich klienty alespoň jednou ročně aktualizovat údaje o doménách ve Whois databázi. Zjistí-li ICANN nějaké nedostatky, potom klientovi poskytne 15 dní na opravu. Jinak hrozí zrušení domény [8]. ICANN provádí kontroly jak automatizovaně, tak i ručně, navíc každý uživatel Internetu si může stěžovat na konkrétní doménu na této adrese [2].

Organizace ICANN problematiku přesnosti databáze Whois průběžně monitoruje. Z auditu za rok 2007 vyplývá, viz tab. 3.2, že nejpočetnější doménou s problémem byla doména `.com`, u které bylo zaregistrovánou 25.136 hlášení, celých 73,87 procenta všech hlášení. Ovšem relativně, vzhledem k počtu 10.000 zaregistrovaných domén si domény `.com` s koeficientem 4,27 vedly průměrně. Nejmenší problémy měly domény `.name` s konkrétními jmény. Je to nejspíše proto, že je jich poměrně malý počet a pro vlastníka domény první úrovně se svým vlastním jménem je prestižní záležitostí udržovat ji vpořádku. Počty hlášení o chybách slouží k orientační informaci o přesnosti databáze Whois.



Obr. 3.3: Výpis z Whois databáze.

3.2.2 Databáze MaxMind

Na Internetu je celá řada geolokačních databází. Z dostupných bych zmínil databáze firmy MaxMind, a to právě pro jejich přesnost. Databáze od MaxMind jsou dostupné jak v komerční formě, tak i ve formě volně ke stáhnutí. Komerční geolokační databáze jsou GeoIP City, GeoIP Country, GeoIP Region, GeoIP Organization, GeoIP ISP, GeoIP Netspeed, GeoIP Domain Name. Také nabízí všechny uvedené databáze shrnuty v jedné databázi – GeoIP Omni Web Service. MaxMind nabízí různé typy licenční politiky: Buď se platí za licenci na jeden server nebo za počet geolokačních dotazů na servery MaxMind [13]. Volně stažitelná databáze jsou zastoupeny databázemi GeoLite Country a GeoLite City pro IPv4 a pro IPv6 [4].

Databáze MaxMind jsou distribuovány buď v binární nebo v textové podobě, kdy se jedná o soubor typu CSV. Volně dostupná databáze GeoLite City se skládá ze dvou souborů GeoLiteCity-Blocks.csv a GeoLiteCity-Location.csv. Postup vyhledávání je následující: V GeoLiteCity-Blocks vyhledám blok adres, do kterého patří moje IP adresa. Adresy zde nejsou v běžném čtyřbajtovém zobrazení. Adresy jsou zde uvedeny jako celé číslo. Z běžného zápisu `AAA.BBB.CCC.DDD` převedu adresu na celé číslo `X` pomocí vzorce (3.1)

$$X = AAA \cdot 256^3 + BBB \cdot 256^2 + CCC \cdot 256 + DDD \tag{3.1}$$

kde X je celé číslo,

AAA.BBB.CCC.DDD je IP adresa.

Tab. 3.2: Přesnost údajů ve Whois databázi, převzato z [8]

TLD	Unikátních hlášení (-)	Unikátních hlášení (%)	Unikátních hlášení na 10,000 registrací*
.com	25.136	73,87	4,27
.net	4.734	13,91	5,47
.info	2.563	7,53	6,77
.biz	311	0,91	1,98
.org	1281	3,76	2,33
.name	4	< 0,01	0,175
Celkem	34.029	100	4,33

* Stav registrací k 30. listopadu 2006.

V GeoLiteCity-Blocks určím pomocí mezí `startIpNum` a `endIpNum` blok s `locId`. Identifikátor bloku `locId` slouží jako klíč do databáze GeoLiteCity-Location, kde vyhledám zemi, region, město a souřadnice daného města.

Zajímavým faktem je, že MaxMind má u svých databází vyjádřenu přesnost. Přesnost pro Evropu je uvedeno v tabulce 3.3.

Tab. 3.3: Přesnost databází MaxMind pro Evropu, převzato z [13]

Země	**Správně určeno (do 40 kilometrů) (%)**	**Nesprávně určeno (%)**	**Město je neznámé (%)**
Austria	73	18	9
Belgium	84	13	3
Bulgaria	75	18	7
Croatia	72	25	3
Czech Republic	82	14	4
Denmark	84	11	5
Finland	66	24	10
France	67	26	7
Germany	78	18	4
Greece	69	29	2
Hungary	70	22	8
Iceland	83	16	1
Ireland	51	21	28
Italy	63	29	8
Luxembourg	87	8	5
Netherlands	85	11	4

Norway	78	18	4
Poland	58	36	6
Portugal	74	21	5
Romania	69	24	7
Slovakia	45	25	30
Spain	72	24	4
Sweden	72	27	1
Switzerland	73	15	12
United Kingdom	72	21	7
United States (pro srovnání)	81	15	4

Přesností databází se zabýval Poese et al. [16]. Odborníci změřili počet bloků a počet unikátních poloh (měst) a porovnávali tento poměr, viz tab. 3.4. Z tohoto porovnání vyšla databáze MaxMind jako databáze s nejlepším poměrem unikátních poloh vůči počtu bloků.

Tab. 3.4: Obecné charakteristiky studovaných databází, převzato z [16].

Databáze	**Bloky**	**Počet unikátních souřadnic**	**Země**	**Města**
Host IP	8.892.291	33.680	238	23.700
IP2Location	6.709.973	17.183	240	13.690
InfoDB	3.539.029	169.209	237	98.143
Maxmind	3.562.204	203.255	244	175.035
Software77	99.134	227	225	0

MaxMind se věnuje ještě další oblasti, a to vyhodnocování podvodů při online platbách. Vytvořili na tuto problematiku databázi, která obsahuje bezpečnostní prvky, zejména se vyhodnocuje tzv. riskScore, to je výsledek bezpečnostního zhodnocení transakce [13].

4. VYTVOŘENÝ PROGRAM NA LOKALIZACI ÚTOČNÍKA

Součástí zadání této diplomové práce je vytvořit aplikaci zjišťující polohu potenciálního útočníka na operační systém. Aplikaci jsem navrhl jako soubor několika programových modulů. Moduly byly realizované v jazyce C, internetová část potom v Javascriptu. Programování aplikace jsem prováděl pod operačním systémem Linux, konkrétně Ubuntu 12.10 Aplikace je kompatibilní se současnými verzemi operačního systému Linux.

Hacker útok zahajuje nejčastěji tzv. skenováním portů, například pomocí programu Nmap. Nmap zjišťuje stav portů cílové stanice, to umožňuje útočníkovi zjistit, jaké síťové aplikace na počítači běží, případně typ a verzi operačního systému.

V rámci řešení diplomových a bakalářských prací vznikla v tomto roce pod vedením pana doc. Komosného skupina, v rámci níž jsme spolupracovali a díky tomu jsme na závěr mohli porovnat výsledky geolokace prováděné různými metodami. Aby bylo vzájemné srovnání možné, bylo nutno geolokaci provádět na stejné množině cílů (datasetu). V rámci vytvořeného datasetu jsme určili skutečnou polohu vybraných serverů, kterou potom porovnáváme se zjištěnou polohou a určujeme chybu měření. Díky tomu bylo nutné programový modul `asimul` upravit tak, aby prováděl útok pouze z dané množiny adres v datasetu.

4.1 Struktura a provedení programu

V předkládané diplomové práci jsem realizoval, v souladu se zadáním, aplikaci, která zjišťuje polohu potenciálního útočníka na operační systém. Aplikaci jsem rozvrhl do tří programových modulů.

Modul `asimul` simuluje útok na operační systém formou skenování portů. Vzhledem k výše uvedeným důvodům je útok realizován výběrem z množiny útočníků z námi vytvořeného datasetu. Modul `adetector` provádí detekci útoku skenováním portů. Modul načítá a zpracovává systémový soubor logu, po zjištění útoku provede geolokaci útočníka dotazem do databáze `GeoLite City` od firmy MaxMind [4]. Databáze je uložena na pevném disku počítače. Zjištěná a skutečná poloha potenciálního útočníka je potom předána modulu webové stránky, kde je následně zobrazena v mapě. Použité mapy jsou načítány ze serveru Google maps a mají tedy celosvětový rozsah, je možné zobrazit libovolnou světovou adresu. V rámci práce jsme se zaměřili pouze na evropské servery.

4.1.1 Modul asimul

Modul `asimul` simuluje útok na operační systém formou skenování portů. Jak již bylo zmíněno, IP adresy útočníků jsou vybírány z námi vytvořeného datasetu.

Na začátku programu je provedeno načtení a kontrola parametrů z příkazového řádku, viz obr. 4.1. Obsluha příkazového řádku je provedena funkcí `ZpracujParamPrikRadky`, která využívá knihovní funkci `getopt`. Tato funkce zpracuje parametry příkazového řádku zadané

v libovolném pořadí a rozlišuje mezi povinnými a volitelnými parametry. Zjistí-li program chybu v zadání z příkazové řádky, vypíše nápovědu a skončí.

Program načte ze souboru (definovaného povinnou volbou `-x`) IP adresy potenciálních útočníků a uloží si je do paměti. Načítání je prováděno funkcí `NactiSeznamIP` s použitím knihovní funkce `fgets`.

Následuje rozhodnutí, zda-li byla v programu `asimul` použita volba `-p?` Pokud ano, tak program ověří, zda se zadaná IP adresa vyskytuje v datasetu, jestliže ano, potom z ní provede útok. Pokud volba `-p` nebyla zadána, pak program přechází do kontinuálního režimu. V tomto režimu v cyklu generuje náhodně IP adresu z datasetu, provede útok a potom usne na předepsanou dobu. Pro tento režim je potřeba program spouštět na pozadí, což se v příkazovém interpretu Bash provádí symbolem & na konci příkazu. Volitelně mohu intenzitu útoku ovlivnit volbou `-n`, která specifikuje počet útočníků v desetiminutovém intervalu. Když tuto volbu nepoužiji, potom bude použito standardní nastavení, tj. 30 útoků za 10 minut.

4.1.2 Modul adetector

Modul `adetector` průběžně detekuje potencionální útočníky, kteří formou skenování portů zahajují útok na operační systém.

Na začátku programu jsou, podobně jako u výše uvedeného modulu `asimul`, zkontrolovány parametry příkazové řádky a případně je vypsána nápověda k použití programu, viz obr. 4.2.

Program pomocí funkce `NactiSeznamIP_seSouradnicemi` načte z datasetu IP adresy útočníků a jejich skutečnou zeměpisnou polohu ze souboru. Funkce využívá knihovní funkci `fgets` pro čtení ze souboru. Z načtených dat je potom zkopírována skutečná poloha útočníka pomocí knihovní funkce `strncpy`. Dále program vstoupí do cyklu, ve kterém je periodicky volána funkce `NactiUtocniky`. Modul `adetector` je potřeba rovněž spouštět na pozadí.

Funkce `NactiUtocniky` provádí čtení ze souboru log (specifikovaným parametrem `-i`), viz obr. 4.3. Pro každý načtený řádek je potom pomocí funkce `AkceptujRadek` rozhodnuto, zda řádek akceptovat, či ne. Přijaty jsou pouze přidané nové řádky, které dosud programem nebyly zpracovány. Program dále vyhledá v paměti skutečnou polohu pro danou IP adresu potenciálního útočníka a zkopíruje ji do struktury aktuálního útočníka. Následně funkce provede dotaz do databáze `GeoLite City [4]`, pomocí programu `geoiplookup` od firmy MaxMind. Zjištěné údaje funkce vypisuje do konzoly pomocí knihovní funkce `printf`. Nakonec funkce zavolá internetový prohlížeč Firefox, přičemž mu předá parametry pro modul webové stránky. Parametry jsou chystány na několika řádcích pomocí knihovní funkce `sprintf`. Samotný internetový prohlížeč je zavolán pomocí knihovní funkce `system`. V prohlížeči je potom v mapě zobrazena zjištěná a skutečná poloha potenciálního útočníka. V kontextové nápovědě je následně možno zjistit parametry útoku včetně vzdálenosti mezi těmito dvěma polohami, což představuje chybu měření.

4.1.3 Webová stránka

Webová stránka (modul `dva_body_na_mapev101.html`) slouží k zobrazení zjištěných informací v mapě. Používám mapové podklady a aplikační rozhraní (API) Google map. Využití Google map má výhodu v tom, že tyto mapy mají celosvětové pokrytí. V rámci našeho datasetu jsme se zabývali pouze evropskými servery.

Rozhraní Google map je realizováno pomocí objektů v Javascriptu. Pro práci s rozhraním jsem vytvořil stránku v HTML doplněnou o segmenty v jazyce Javascript. Při spuštění webové stránky se nejprve načte rozhraní Google map:

```
<script type="text/javascript" src=https://maps.googleapis.com/maps/api/js?libraries=geometry&key=KLIC&sensor=true> </script>
```

Kde `KLIC` je unikátní klíč vytvořený v uživatelském účtu Google. V položce `libraries` specifikuji knihovnu `geometry`, kterou využívám k vypočtení vzdálenosti mezi zjištěnou a skutečnou polohou.

Výkonná část programu je realizována funkcí `initialize`, která je zavolána po načtení stránky. Dále se tedy budu zabývat touto funkcí. Předané parametry zpracuji pomocí funkce `split` takto:

```
var parametry = top.location.href.split('?');
```

Bod v mapě vytvořím pomocí funkce `LatLng`:

```
var m1 = new google.maps.LatLng(lat, lng);
```

Parametry mapy reprezentuje struktura:

```
var mapOptions = {
  zoom: 4,
  center: m1
  mapTypeId: google.maps.MapTypeId.ROADMAP};
```

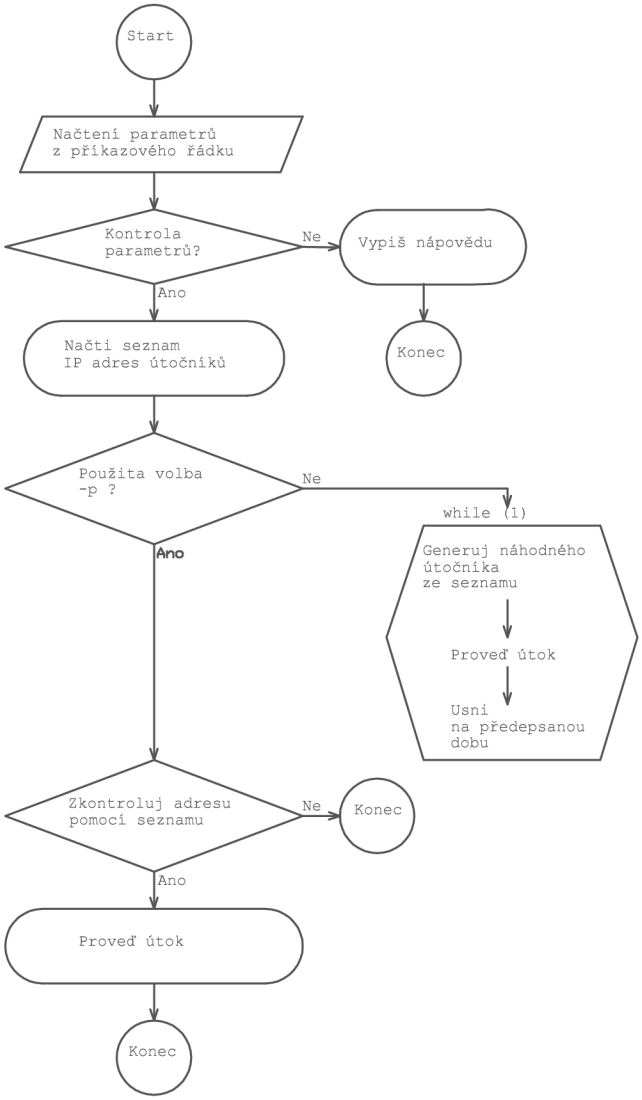
Mapu potom vytvořím v elementu `<div>` s názvem `map_canvas` příkazem:

```
var map = new google.maps.Map(document.getElementById('map_canvas'), mapOptions);
```

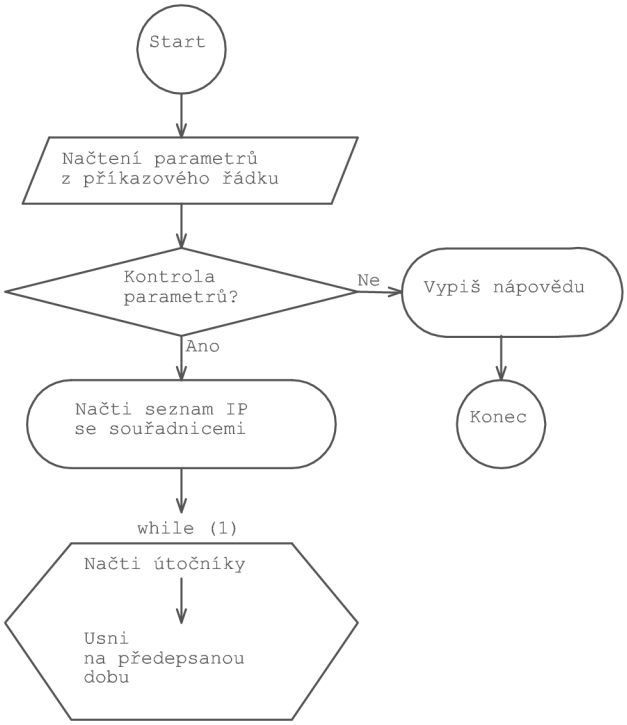
Vzdálenost mezi dvěma místy vypočtu:

```
var vzdalenost = google.maps.geometry.spherical.computeDistanceBetween (m1,m2)
```

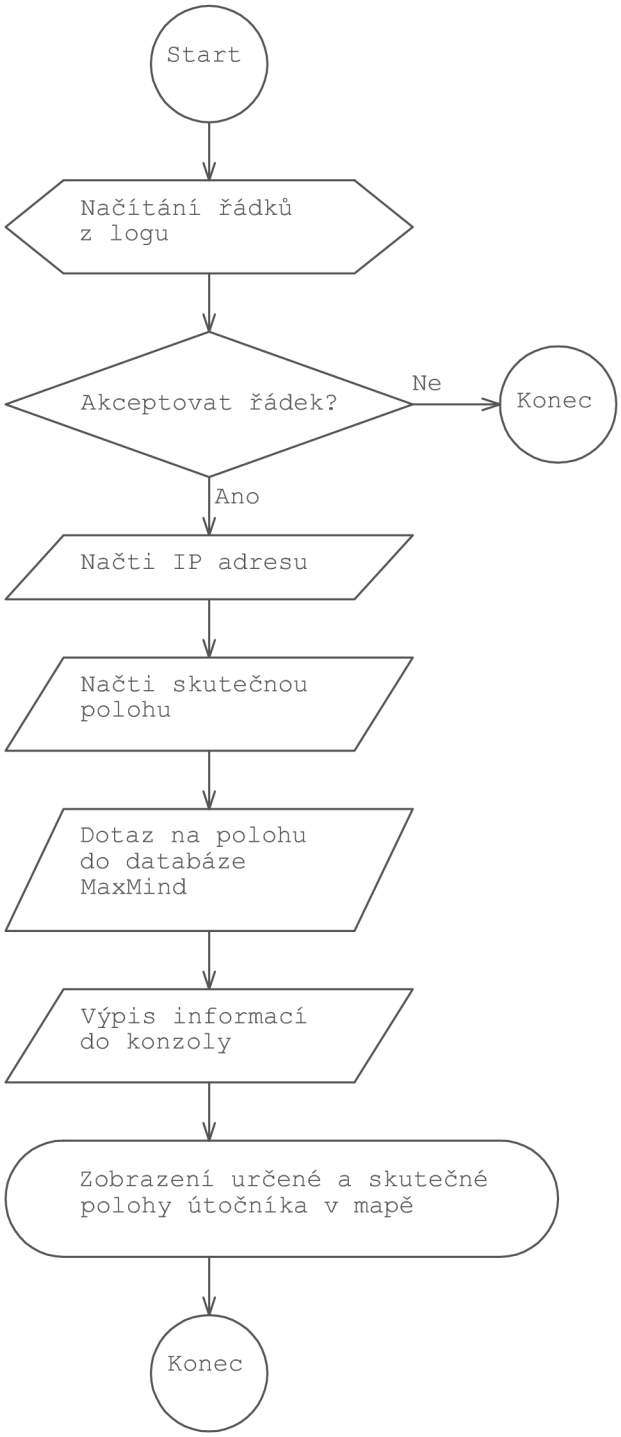
Webová stránka zobrazí zjištěnou a skutečnou polohu v mapě a spočítá vzdálenost mezi těmito dvěma polohami, viz. obr. 4.5.



Obr. 4.1: Vývojový diagram modulu asimul.



Obr. 4.2: Vývojový diagram modulu adetector.



Obr. 4.3: Vývojový diagram funkce NactiUtocniky (modul adetector).

4.2 Výstupy z programu

V této podkapitole bych chtěl přehledně shrnout výstupy a mezivýstupy programových modulů `asimul`, `adetector` a modulu `webové stránky`. Výstupy programu jsou pro uživatele to nejdůležitější. Předkládaný program kromě textových výstupů nabízí i grafické výstupy v podobě mapy.

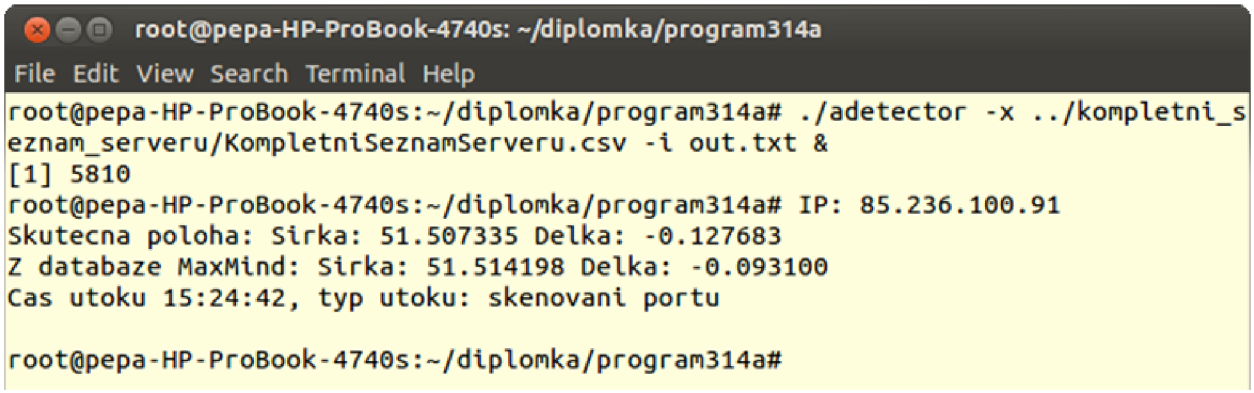
Modul `asimul` provádí do výstupního log souboru následující otisk, který je shodný s otiskem zapisovaným programem `scanlogd`:

```
Apr 16 15:24:42 pepa_desktop scanlogd: 85.236.100.91 to 127.0.0.1 ports …
```

Log soubor je následně periodicky zpracováván na pozadí běžícím modulem `adetector`. Pro pohodlnou práci s daty jsem si definoval strukturu `struct utocnik`, která obsahuje všechny sledované údaje, které je potřeba o potencionálním útočníkovi evidovat:

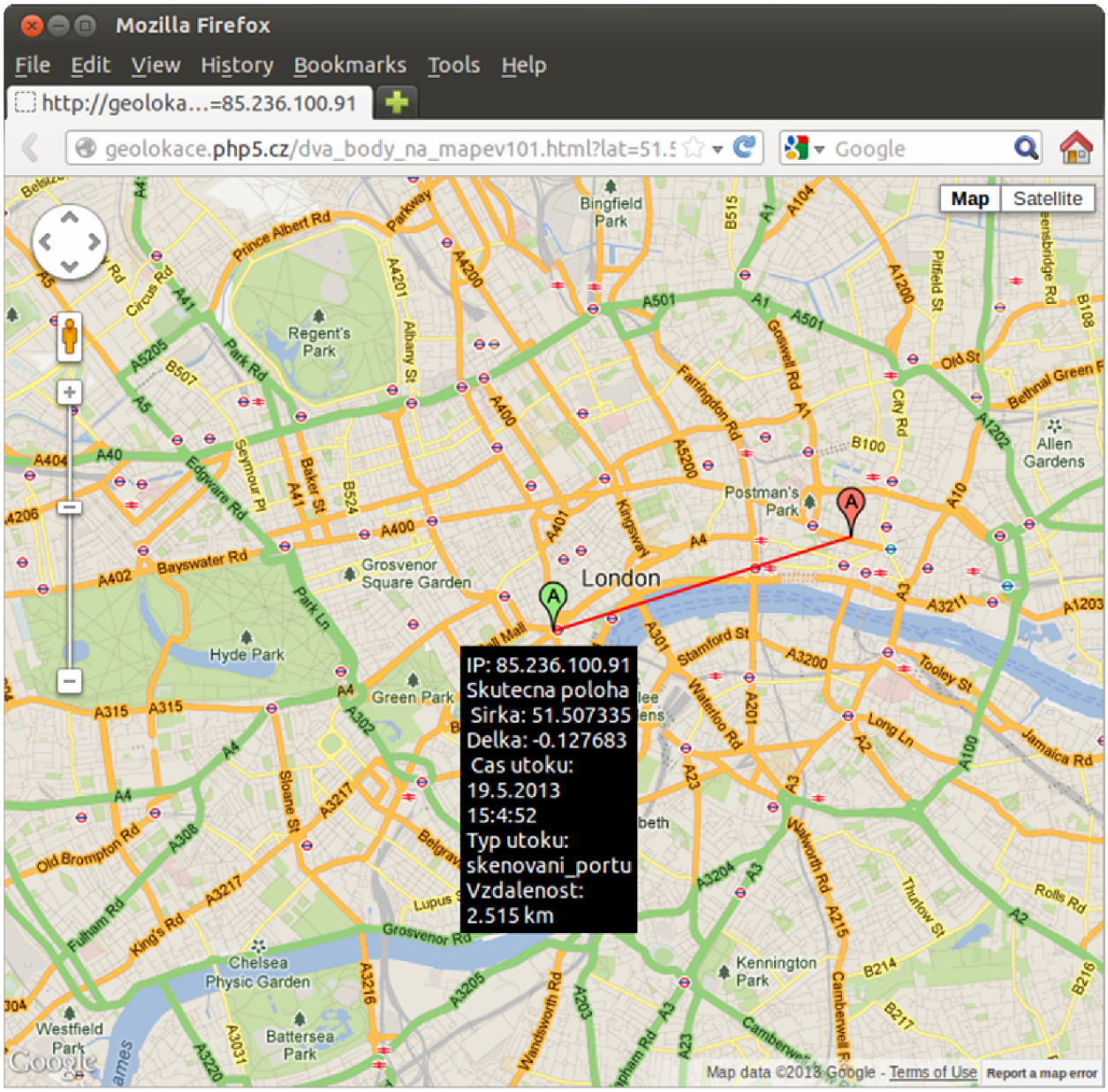
```
struct utocnik {
char IPadresa[IP_ADDR_LEN]; /* IP adresa ABC.DEF.GHI.JKL */
char SirkaMM[3*STD_LEN]; /* Zemepisna sirka z databaze MaxMind */
char DelkaMM[3*STD_LEN]; /* Zemepisna delka z databaze MaxMind */
char SirkaSkut[3*STD_LEN]; /* Zemepisna sirka skutecna */
char DelkaSkut[3*STD_LEN]; /* Zemepisna delka skutecna */
struct tm CasUtoku; /* Cas utoku */
};
```

Uvedená struktura zahrnuje všechny potřebné údaje o útočníkovi. Tyto údaje jsou následně vypsány do konzoly, viz. obr. 4.4.



Obr. 4.4: Textový výstup z modulu `adetector`.

Údaje uložené ve struktuře `struct utocnik` jsou zformátovány, předány modulu `webové stránky` a zobrazeny v internetovém prohlížeči Firefox, viz. obr. 4.5.



Obr. 4.5: Mapový výstup modulu `webová stránka`.

Uvedená mapa obsahuje standardní funkcionality Google map, jako je možnost posunování, zoom, StreetView apod. Zoom lze provést buď pomocí standardních tlačítek nebo kliknutím na značku.

U zelené nebo červené značky lze vyvolat kontextovou nápovědu, která obsahuje tyto údaje: IP adresa útočníka, skutečná poloha případně poloha určená z databáze MaxMind, čas útoku, typ útoku, vzdálenost mezi zelenou a červenou značkou. Zelená značka udává skutečnou polohu útočníka a červená zjištěnou polohu útočníka. Rozdíl mezi těmito dvěma polohami (v mapě vyznačen červenou spojnicí) udává chybu v kilometrech, což představuje chybu měření.

5. MĚŘENÍ A DISKUZE

Určování zeměpisné polohy potenciálního útočníka v Internetu je možné provádět dvěma způsoby, a to aktivními metodami měřícími zpoždění v síti anebo pasivními metodami, tj. dotazy do databáze. V předkládané diplomové práci jsem vytvořil aplikaci, která z veřejně dostupné poziční databáze zjišťuje polohu potenciálního útočníka na operační systém. Rozsah IP adres zpracovávaných aplikací byl po dohodě s vedoucím práce omezen na konkrétní dataset, obsahující reprezentativní evropské servery.

V tomto roce vznikla pod vedením pana doc. Komosného skupina studentů zpracovávajících své diplomové a bakalářské práce zabývající se problematikou geolokace. Díky tomu, že jsme všichni používali stejný dataset evropských serverů, je možno objektivně porovnat výsledky různých metod, jak aktivních, tak pasivních metod geolokace. V práci tedy uvádím a srovnávám výsledky své s výsledky kolegů. Použití výsledků kolegů uvádím s jejich souhlasem. V předkládané diplomové práci jsou porovnány výsledky těchto metod:

- Geolokace pomocí databáze `GeoLite City` od firmy MaxMind,
- Geolokace pomocí databáze Whois (autor Jan Henek),
- Geolokace pomocí doménových jmen (autor Ondřej Jelínek),
- Geolokace metodou Constraint-Based Geolocation (CBG, autor Bc. Michael Horák).

Tab. 5.1: Výsledky určení polohy stanice pomocí databáze `GeoLite City` od firmy MaxMind.

IP adresa	Místo	Chyba (km)
81.2.194.154	Habry – Česká republika	0
91.235.52.167	Slovensko	0
193.136.163.66	Lisabon - Portugalsko	0
77.47.133.22	Kijev - Ukrajina	0
129.27.201.245	Graz - Rakousko	0,301
81.201.56.141	Plzeň - Česká republika	0,469
131.175.187.11	Milán - Itálie	0,475
89.215.114.195	Plovdiv - Bulharsko	0,687
130.59.10.36	Zurych - Švýcarsko	0,844

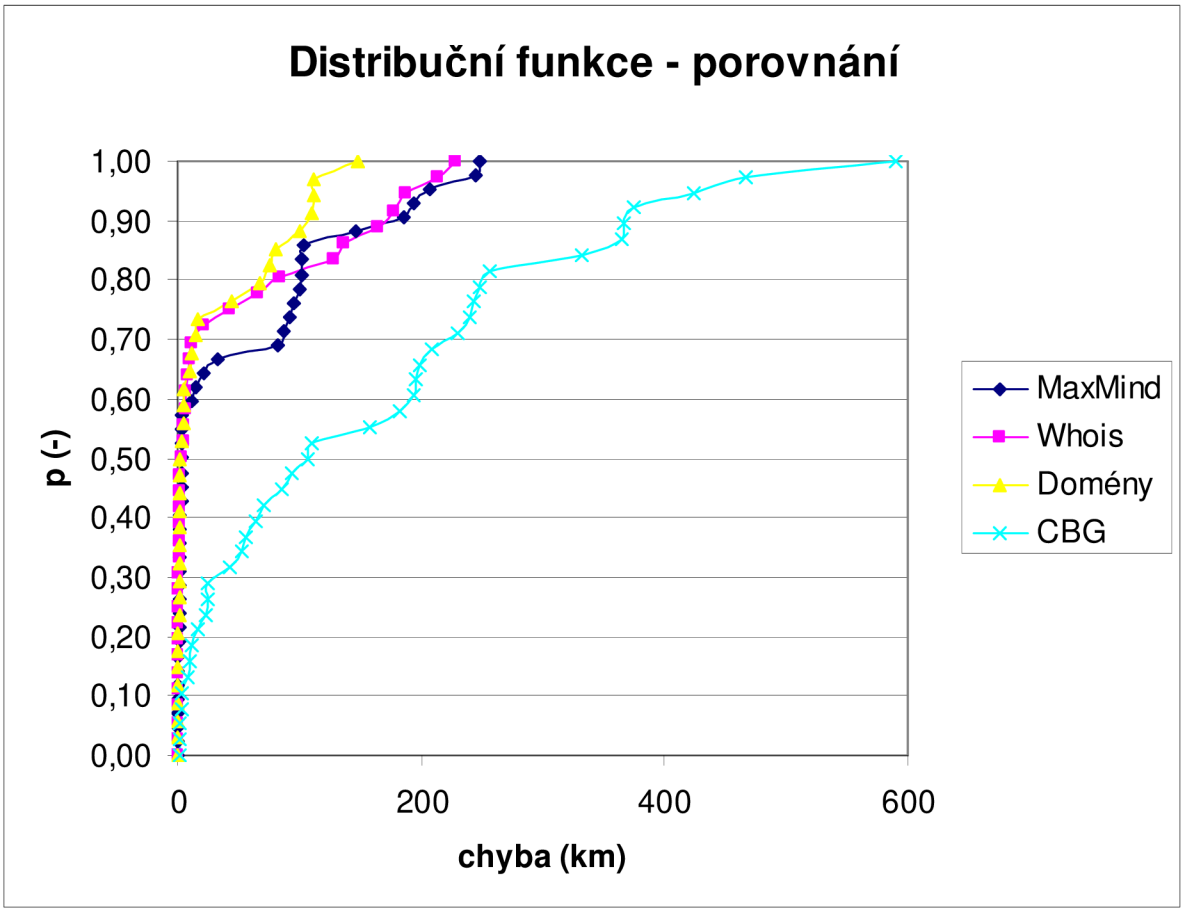
91.90.160.3	Wroclaw - Polsko	0,952
193.166.3.2	Espoo - Finsko	1,386
94.136.136.2	Michlovce - Slovensko	1,510
85.11.157.54	Sofia - Bulharsko	1,655
212.87.14.41	Varšava - Polsko	1,690
132.252.181.87	Essen - Německo	1,713
130.235.209.220	Lund - Švédsko	1,831
130.251.19.2	Janov – Itálie	1,923
80.72.40.110	Varšava - Polsko	2,412
85.236.100.91	Londýn – Anglie	2,515
213.249.64.165	Amsterdam – Nizozemí	2,685
194.146.252.199	Krakow – Polsko	2,898
152.66.115.224	Budapešť – Maďarsko	2,904
130.239.141.10	Umea – Švédsko	2,991
129.240.8.200	Oslo – Norsko	3,028
217.79.128.22	Budapešť – Maďarsko	3,205
93.115.207.238	Jilava – Rumunsko	11,207
109.70.148.245	Londýn – Anglie	15,539
88.190.22.159	Draveil – Francie	20,918
213.153.32.170	Salzburg – Rakousko	33,672
212.67.73.150	Praha – Česká republika	81,800
77.75.76.3	Praha – Česká republika	86,549
91.82.84.185	Budapešť – Maďarsko	91,281

77.251.170.54	Heusden – Nizozemí	95,071
89.216.2.122	Bělehrad – Srbsko	100,859
5.9.59.165	Frankfurt – Německo	101,493
77.93.192.144	Brno – Česká republika	102,314
176.9.55.42	Německo	103,167
109.70.149.49	Stroud – Anglie	146,852
62.65.173.6	Bratislava – Slovensko	186,061
95.77.94.89	Bukurešť – Rumunsko	194,032
212.200.163.178	Kikinda – Srbsko	207,848
78.46.90.47	Gunzenhausen – Německo	244,447
213.94.75.9	Vídeň – Rakousko	247,433

Tabulka 5.1 udává lokaci serverů u nichž byla určována poloha pomocí dotazů do databáze MaxMind. Zjištěná poloha byla potom porovnána se skutečnou polohou a byla určena chyba měření. Servery jsou řazeny vzestupně podle této chyby. Obrázek 5.1 zobrazuje graficky tuto závislost, v obrázku jsou porovnány také údaje od kolegů. Ze zobrazených dat lze usuzovat na přesnost jednotlivých metod.

Moje metoda přístupu ke geolokaci spočívala ve využití databáze `GeoLite City` od firmy MaxMind. Tato databáze se dá volně stáhnout na adrese [4]. Při analýze zpracovaných dat se vyskytly dva problémy: data s určenou minimální chybou a data s relativně velkými chybami. Z analýzy jsem vypustil dva servery, jejichž poloha byla detekována mimo Evropu.

Zvýšenou pozornost vyžadují i servery, které mají minimální, tedy de facto nulovou, chybu měření. Samozřejmě jedná se o ideální měření, pokud dostanu výsledky s minimální chybou. Tyto výjimečně přesné hodnoty mohou být způsobeny díky metodě přístupu: Jedním z prvních úkolů, které jsme od vedoucího našich bakalářských a diplomových prací dostali, bylo nalezení dostatečného množství evropských serverů, u nichž bude známa jejich poloha. Práci jsme si mezi sebe rozdělili. Výsledkem byl dataset evropských serverů. Data v tomto datasetu jsou ovšem zatížena chybou. Při lokalizaci serveru jeden student použil databázi Whois, jiný databázi MaxMind a jiný zase Google mapy. Nulová chyba je tedy u serverů, jejichž skutečná poloha byla nejspíše určena pomocí databáze MaxMind, a tedy dotazem do stejné databáze dostávám identické hodnoty, tedy nulovou chybu geolokace.



Obr. 5.1: Porovnání distribučních funkcí u jednotlivých metod.

Tab. 5.2: Statistické porovnání použitých metod geolokace.

Metoda	Aritmetický průměr (km)	Medián (km)
Databáze MaxMind GeoLite City	49,038	2,904
Databáze Whois	40,965	2,641
Doménové názvy	26,842	2,360
Constraint Based Geolocation	156,093	106,432

Problémem jsou ovšem také servery, u nichž byla vzdálenost větší než určitá tolerance. Maximální chybu jsem nastavil na 300 km, protože se domnívám, že tato vzdálenost stačí k určení státu, ve kterém je daný server provozován. Z naměřených dat jsem potom vypustil ještě dva servery, které tuto podmínku nesplňovaly.

Porovnáním statistických charakteristik: aritmetického průměru a mediánu zjišťujeme, že nejpřesněji se podařilo geolokalizovat stanice kolegovi Ondřeji Jelínkovi metodou analýzy doménových názvů. Zajímavější z pohledu této práce je srovnání mých naměřených dat z databáze `GeoLite City` od firmy MaxMind s databází Whois (autor Jan Henek). Určení polohy pomocí obou geolokačních databází vykazuje podobné výsledky.

Za povšimnutí stojí i malé chyby, řádově stovky metrů nebo kilometr, které byly způsobeny určováním polohy pomocí různých databází (MaxMind, Whois, Google mapy, poloha udaná správcem serveru). Tyto malé chyby by se daly eliminovat určením polohy na město a polohu (středu) tohoto města potom určovat jednotně, například pomocí Google map.

Nejzajímavější je ovšem porovnání pasivních a aktivních metod. Porovnáním hodnot v tab. 5.2 zjišťujeme, že výsledky metody CBG, která se řádí mezi aktivní metody, jsou mezi porovnávanými hodnotami nejhorší. Je to z toho důvodu, že aktivní metody jsou složitější a náročnější na počet provedených měření, z nichž každé je zatíženo nějakou chybou, a tedy chyby se kumulují. Konkrétně u metody CBG záleží také na způsobu hledání cíle, kdy algoritmus zpravidla převádí oblast, kde se nachází cíl, na n-úhelník. Zatímco pasivní databázové metody jsou rychlejší a jednodušší na implementaci, jejich problém spočívá v nákladném budování databází. Z naměřených hodnot ovšem vyplývá, že pasivní databázové metody jsou stále perspektivní.

6. ZÁVĚR

Ve své diplomové práci jsem se zabýval problematikou určení fyzické polohy potenciálního útočníka na operační systém. Určování fyzické polohy stanice v Internetu lze principiálně dvěma způsoby: pomocí aktivních nebo pasivních metod. Aktivní metody spočívají v měření zpoždění v Internetu. Pasivní metody využívají ke geolokaci údaje o internetových adresách uložené v databázi. Zaměřil jsem se podle zadání na pasivní metody, tj. využití databází. Z dostupných geolokačních databází jsem volil volně dostupnou databázi `GeoLite City` od firmy MaxMind. Databáze firmy MaxMind mají totiž velmi dobrou přesnost. Rovněž firma MaxMind nabízí databáze ke stažení a umožňuje práci s nimi na lokálním počítači.

Zabýval jsem se principy útoků na operační systém. Z možných typů útoků jsem se zaměřil na častou formou útoku pomocí skenování portů, pracoval jsem s programem Nmap. Naprogramoval jsem aplikaci, která simuluje útočníka zahajujícího útok na operační systém (program `asimul`). Program `asimul` vybírá reálné IP adresy z konkrétního datasetu evropských serverů, aby bylo možno provést vzájemné srovnání, viz. kapitola 5. Dále jsem navrhl a ověřil aplikaci (program `adetector`) pro detekci útoku a určení geografické polohy útočníka na operační systém. `Adetector` pracuje s údaji z volně dostupné databáze `GeoLite City` od firmy MaxMind. Na závěr je skutečná poloha stanice z datasetu a zjištěná poloha stanice předána modulu webové stránky, který zajistí jejich přehledné zobrazení v mapě.

Závěrečná kapitola shrnuje dosažené výsledky. Díky tomu, že jsme, po dohodě s vedoucím, používali všichni stejný dataset evropských serverů, je v této kapitole i vzájemné porovnání pasivních a aktivních metod. Kolega Jan Henek používal pro geolokaci databázi Whois. Při porovnání s jeho daty (aritmetický průměr chyby: 40,965 km; medián chyby: 2,641 km) jsem zjistil, že mnou použitá databáze `GeoLite City` od firmy MaxMind udávala podobné výsledky (aritmetický průměr chyby: 49,038 km; medián chyby: 2,904 km). Zajímavé bylo rovněž porovnání našich výsledků s kolegou Bc. Michaelem Horákem, který se zabýval aktivní metodou CBG. Kolega Horák naměřil aritmetický průměr chyby: 156,093 km; medián chyby: 106,432 km). Porovnáním těchto údajů s údaji z pasivních metod (databáze Whois, databáze MaxMind) vychází, co se týče přesnosti, že databázové údaje mají lepší přesnost. Rovněž rychlost a jednoduchost práce s databázemi je pro jejich použití výhodná. Největší nevýhodou databázových metod je nákladnost a údržba samotných geolokačních databází.

Diplomová práce se zabývala velmi aktuální tématikou zvýšení bezpečnosti díky možnostem lokalizace útočníka na operační systém a podle výše uvedeného cíl splnila.

SEZNAM POUŽITÝCH ZDROJŮ

[1] BALEJ, J.; KOMOSNÝ, D. Zdroje zpoždění při komunikaci v Internetu. In *Elektrorevue,* 2010/42, 21.6.2010. ISSN 1213-1539. Dostupné z URL: <http://www.elektrorevue.cz>

[2] *Contractual Compliance Complaints – Whois Inaccuracy Form.* [online]. [cit. 27.4.2013]. Dostupné z URL: <http://www.icann.org/en/resources/compliance/complaints/whois/inaccuracy-form>

[3] GEIGER, J. *Pohled do jádra: Jak silná jsou Windows.* [online]. [cit.8.12.2012]. Chip Online, 2008. Dostupné z URL: <http://earchiv.chip.cz/cs/earchiv/vydani/r-2008/pohled-do-jadra.html>

[4] *GeoLite Free Downloadable Databases.* [online]. [cit. 1.3.2013]. Dostupné z URL: <dev.maxmind.com/geoip/geolite>

[5] GUEYE, B. et al. *Constraint-Based Geolocation of Internet Hosts.* [online]. [cit. 9.5.2013]. Dostupné z URL:

<http://www.cs.bu.edu/fac/crovella/paper-archive/imc04-geolocation-full.pdf>

[6] HATCH, B.; LEE, J.; KURTZ, G. *Linux Hackerské Útoky: Bezpečnost Linuxu – tajemství a řešení.* Z anglického originálu Hacking Linux Exposed: Linux Security Secrets & Solutions přeložili Kuklínek, J.; Paulíček, L. Praha: SoftPress, 2002. 576 s. ISBN 80-86497-17-8.

[7] *IANA - Number Resources.* [online]. [cit. 10.5.2012]. Dostupné z URL: <http://iana.org/numbers>

[8] *ICANN'S Whois Data Accuracy and Availability Program: Description of Prior Efforts and New Compliance Initiatives.* [online]. 27.4.2007. [cit. 27.4.2013]. Dostupné z URL: <http://www.icann.org/whois/whois-data-accuracy-program-27apr07.pdf>

[9] *Intel SSE4 Programming Reference.* [online]. [cit.8.12.2012]. Intel Corporation, Denver, USA, July 2007. Reference Number: D91561-003. Dostupné z URL: <software.intel.com/file/18187/>

[10] JELÍNEK, L. *Jádro systému Linux. Kompletní průvodce programátora.* Brno: Computer Press, 2008. 686 s. ISBN 978-80-251-2084-2.

[11] LYON, G. F. *Nmap Network Scanning. The Official Nmap Project Guide to Network Discovery and Security Scanning* [online]. [cit. 3.12.2012]. Insecure.Com, 2011. ISBN 978-0-9799587-1-7. Dostupné z URL: <http://nmap.org/book/toc.html>

[12] MACICH, J. *Bezpečnost Windows 8: Microsoft vylepšil nástroje pro rodiče i ochranu citlivých dat.* [online]. [cit.8.12.2012]. Portál Lupa.cz, 30.10.2012. Dostupné z URL:

<http://www.lupa.cz/clanky/bezpecnost-windows-8-microsoft-vylepsil-nastroje-pro-rodice-i-ochranu-citlivych-dat/>

[13] *MaxMind – IP Geolocation and Online Fraud Prevention.* [online]. [cit.8.12.2012]. Maxmind, Waltham, USA, 2012. Dostupné z URL: <http://www.maxmind.com>

[14] NOSKA, M. *MinWin – revoluční jádro pro Windows.* [online]. [cit.8.12.2012]. Computerworld, 21.4.2008. Dostupné z URL: <http://computerworld.cz/vyvoj/minwin-revolucni-jadro-pro-windows-1341>

[15] PADMANABHAN, V., N.; SUBRAMANIAN, L. An Investigation of Geographic Mapping Techniques for Internet Hosts. In *Proceeding of SIGCOMM'01,* San Diego, Kalifornia, USA, 2001. s.173-185. Copyright ACM 1-58113-411-8/01/0008. [online]. [cit. 20.4.2013]. Dostupné z URL: <http://conferences.sigcomm.org/sigcomm/2001/p14-pabmanabhan.pdf>

[16] POESE, I.; UHLIG, S.; DONNET B.; KAAFAR, M. A.; GUEYE, B. *IP Geolocation Databases: Unreliable?* [online]. [cit.8.12.2012]. Technical University Berlin, Faculty of Electrical Engineering and Informatics, Technical Report 2011-03, February 2011. ISSN 1436-9915. Dostupné z URL: <http://www.net.t-labs.tu-berlin.de/papers/PKDGU-IGDU-11.pdf>

[17] VERNER L.; KOMOSNÝ D. Geolokace síťových zařízení v internetových sítích. In *Elektrorevue,* 2011. 17.6.2011. ISSN 1213-1539. Dostupné z URL: <http://www.elektrorevue.cz>

[18] *Welcome to ICANN!* [online]. [cit. 8.12.2012]. ICANN.org. Dostupné z URL: <http://www.icann.org/en/about/welcome>

SEZNAM POUŽITÝCH ZKRATEK, VELIČIN A SYMBOLŮ

API	Application Programming Interface
ARP protokol	Address Resolution Protocol
CBG	Constraint-Based Geolocation
CIDR	Classless Inter-Domain Routing
CSV soubor	Comma Separated Values
HTML	Hypertext Markup Language
IANA	Internet Assigned Numbers Authority
ICANN	Internet Corporation for Assigned Names and Numbers
ICMP protokol	Internet Control Message Protocol
IP protokol	Internet Protocol
ISP	Internet Service Provider
LAN	Local Area Network
NFS	Network File System
NMAP	Network Maper
NSE skript	Nmap Scripting Engine skript
OS	Operating System
PIN	Personal Identification Number
RIR	Regional Internet Registry
RPC	Remote Procedure Call
RTT	Round Trip Time
SCTP protokol	Stream Control Transmission Protocol
SNMP protokol	Simple Network Management Protocol
TCP protokol	Trasmission Control Protocol
TTL	Time to Live
UDP protokol	User Datagram Protocol
WDRP	Whois Data Reminder Policy

SEZNAM PŘÍLOH